

EXHIBIT

RX-241

## Advance Information

This document contains information on a product under development. The parametric information contains target parameters that are subject to change.

## Distinguishing Features

- 110 MHz Pipelined Operation
- VGA Compatible
- Mixed Video and Graphics
- 32-bit Graphics and 32-bit Video Pixel Ports
- YCrCb-to-RGB Conversion
- YCrCb 4:2:2 and 2:1:1 Interpolation
- Uses Brooktree's VideoCache™ Technology
- Horizontal Video Up-Scaling
- 64 x 64 x 2 Cursor
- VRAM Shift Clock Support
- Enables DRAM-Based Motion Video Systems
- Programmable Video Extents
- Programmable Color Keying
- Onboard TTL Clock Doubler
- Three 256 x 8 Color Palette RAMs
- Simplifies Integration of Video into Microsoft Windows™
- 3 x 24 Cursor Color Palette
- Standard MPU Interface
- Power-Down Mode
- Directly Implements Brooktree's VideoCache™ Connector
- 160-Pin PQFP Package

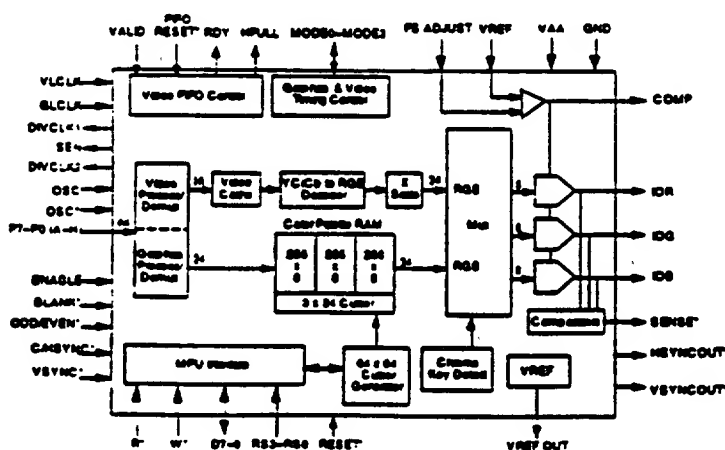
## Applications

- Video Decompression Acceleration
- Multimedia Workstations
- High-Resolution Graphics
- Desktop Video

## Related Products

- Bt812 Video Decoder
- Bt858 Video Encoder
- Bt895 Video Controller
- Bt81295 Personal Media Adapter

## Functional Block Diagram



Brooktree Corporation • 9868 Scranton Road • San Diego, CA 92121-3707  
 (619) 452-7580 • (800) 2BT-APPS • TLX. 383 596 • FAX: (619) 452-1249  
 Internet: apps@brooktree.com  
 L885001 Rev. D

# Bt885

## 110 MHz Monolithic CMOS Video CacheDAC™

## Product Description

The Bt885 is designed specifically for dual or unified frame buffer multimedia subsystems. A dedicated video port accepts a CCIR601 YCrCb or RGB data stream and allows on-screen switching on a pixel-by-pixel basis. Mixing occurs within programmable video extents based on a flexible color key mechanism. Bt885 is intended to replace multiple RAMDAC™-based multimedia subsystems. The Bt885 register set is VGA compatible.

The Bt885 can accelerate video decompression and work with the Bt812 decoder chip using programmable interpolation to pixel multiply by 1, 2, or 4 for CCIR601 4:2:2, 2:1:1, and 1:0.5:0.5 formats. This allows the video data to mix with the graphics data at the same rate.

Brooktree's 800-byte VideoCache™ FIFO enables asynchronous delivery of graphics and video, easing system bandwidth requirements for video transfer, and allowing efficient use of system memory. Non-integer scaling permits arbitrary video window sizing.

The 64 x 64 x 2 bit cursor has its own palette and has priority over the video or graphics. The cursor operates in three modes: Microsoft Windows™, three-color, and X Windows.

The Bt885 supports independent 32-bit graphics and 32-bit video pixel ports and is compatible with both VRAM- and DRAM-based video subsystems.

The Bt885 generates RS-343A-compatible video signals into a doubly terminated 75 Ω load.

# Brooktree®

AT1019E95

Copyright © 1993 Brooktree Corporation. All rights reserved.  
Print date: 05/03/94.

Brooktree reserves the right to make changes to its products or specifications to improve performance, reliability, or manufacturability. Information furnished by Brooktree Corporation is believed to be accurate and reliable. However, no responsibility is assumed by Brooktree Corporation for its use; nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by its implication or otherwise under any patent or patent rights of Brooktree Corporation.

**Specifications are subject to change without notice.**

PRINTED IN THE UNITED STATES OF AMERICA

ATI019596



## Table Of Contents

---

<b>Circuit Description .....</b>	<b>1</b>
MPU Interface .....	1
Hardware Reset Condition .....	1
Writing Color Palette RAM Data .....	1
Reading Color Palette RAM Data .....	1
Writing Cursor and Overscan Color Data .....	3
Reading Cursor Color Data .....	3
Extended Register Mechanism .....	3
Writing Color Key Color Data .....	3
Reading Color Key Color Data .....	3
Writing Color Key Mask Data .....	3
Reading Color Key Mask Data .....	3
Additional Information .....	3
Accessing the Cursor RAM Array .....	4
6-Bit/8-Bit Operation .....	4
Power-Down Mode .....	4
Pixel Clock Selection .....	4
Frame Buffer Pixel Port Interface .....	5
Video Port Clocking .....	5
VideoCache™ FIFO Operation .....	6
Loading VideoCache™ FIFO .....	6
Unloading VideoCache™ FIFO .....	6
HFULL .....	6
VideoCache™ Reset .....	6
DIVCLK1 / DIVCLK2 .....	6
SEN .....	6
MODE0 .....	7
Video Window Operation .....	7
Video Scaling Operation .....	7
Horizontal Scaling .....	8
Color Key Operation .....	8
Example 1 .....	9
Example 2 .....	9
Example 3 .....	9

YCrCb-to-RGB Matrix .....	9
Modes of Operation—Graphics .....	9
4-Bits/Pixel Operation (8:1 MUX) .....	9
8-Bits/Pixel Operation (4:1 MUX) .....	9
8-Bits/Pixel Operation (2:1 MUX) .....	9
8-Bits/Pixel Operation (1:1 MUX) .....	9
16-Bits/Pixel Operation (2:1 MUX) .....	9
16-Bits/Pixel Operation (1:1 MUX) .....	10
24-Bits/Pixel Operation (1:1 MUX) .....	10
Pixel Read Mask Register .....	10
Modes of Operation—Video .....	13
YCrCb 1:0.5:0.5 Operation (4 Bytes/8 Pixels) .....	13
YCrCb 1:0.5:0.5 Operation (2 Bytes/4 Pixels) .....	13
YCrCb 1:0.5:0.5 Operation (1 Byte/2 Pixels) .....	13
YYCrCb 2:1:1 Operation (4 Bytes/4 Pixels) .....	14
YCrCb 2:1:1 Operation (2 Bytes/2 Pixels) .....	14
YCrCb 2:1:1 Operation (1 Byte/1 Pixel) .....	14
YCrCb 4:2:2 Operation (4 Bytes/2 Pixels) .....	15
YCrCb 4:2:2 Operation (2 Bytes/1 Pixels) .....	15
16-Bits/Pixel 5:5:5 Operation (2:1 MUX) .....	15
16-Bits/Pixel 5:5:5 Operation (1:1 MUX) .....	15
16-Bits/Pixel 5:6:5 Operation (2:1 MUX) .....	16
16-Bits/Pixel 5:6:5 Operation (1:1 MUX) .....	16
24-Bits/Pixel Operation (1:1 MUX) .....	16
DAC Values in 16-Bits/Pixel Video Modes .....	18
Cursor Operation .....	19
Cursor Color Support .....	21
Highlight Logic .....	21
Video Generation .....	21
SENSE* Output .....	21
Command Register 0 .....	25

## Internal Registers .....

Command Register 1 .....	26
Command Register 2 .....	28
Accessing the Extended Registers .....	29
Command Register 3 .....	30
Command Register 4 .....	31
Pixel Read Mask Register .....	32
Status Registers 1 and 2 .....	32
Video Window XSTART .....	33

Video Window YSTART .....	33
Video Window XWIDTH .....	33
Video Window YHEIGHT .....	33
XSCALEINIT (Low and High) .....	33
XSCALEINC (Low and High) .....	33
Serial Clock Enable Start (Horizontal) .....	33
Serial Clock Enable Duration (Horizontal) .....	33
DIVCLK1 and DIVCLK2 Rate .....	33
VideoCache™ FIFO Size .....	33
Cursor (x,y) Registers .....	34

## Pin Description..... 36

PC Board Considerations .....	40
Component Placement .....	40
Ground Planes .....	40
Power Planes .....	40
Device Decoupling .....	40
Power Supply Decoupling .....	40
COMP Decoupling .....	40
VREF Decoupling .....	40

## PC Board Layout Considerations..... 40

Digital Signal Interconnect .....	41
TTL Clock Interfacing .....	41
Differential Clock Interfacing .....	41
MPU Control Signal Interfacing .....	41
Analog Signal Interconnect .....	41
Analog Output Protection .....	42
Using Multiple Devices .....	44
ESD and Latchup Considerations .....	44
Sleep Operation .....	44



**THE**

vi

## List of Tables

Table 1.	Control Input Truth Table (RS3 = MSB, RS0 = LSB).....	1
Table 2.	Address Register Operation and Autoincrementing.....	5
Table 3.	5:5:5 RGB Graphics Color Format for Both 2:1 and 1:1 Multiplexing Modes.....	11
Table 4.	5:6:5 RGB Graphics Color Format for Both 2:1 and 1:1 Multiplexing Modes.....	11
Table 5.	Graphics Pixel Index Masking.....	11
Table 6a.	24-Bits/Pixel Graphics RGB Color Format (CR4_1, 40 = 00) for 1:1 MUX Modes.....	12
Table 6b.	24-Bits/Pixel Graphics BRG Color Format (CR4_1, 40 = 01).....	12
Table 6c.	24-Bits/Pixel Graphics BRG Color Format (CR4_1, 40 = 10).....	12
Table 7.	CCIR601 1:0.5:0.5 Video Format.....	17
Table 8.	CIR601 2:1:1 Video Format.....	17
Table 9.	CCIR601 4:2:2 Video Format.....	18
Table 10.	Pseudo Code for Bt885 Monitor Connection.....	20
Table 11.	Cursor Color Modes.....	22
Table 12.	Video Output Truth Table (SETUP = 7.5 IRE).....	23
Table 13.	Video Output Truth Table (SETUP = 0 IRE).....	24
Table 14.	Modes of Operation (Video Pixel Port Configuration).....	26
Table 15.	Modes of Operation (Graphic Pixel Port Configuration).....	27
Table 16.	Extended Registers Address Map (RS3-RS0 = 1010).....	29
Table 17.	Register Values on Reset.....	35

## Circuit Description

## MPU Interface

As illustrated in the detailed block diagram (Figure 1), a standard MPU bus interface is supported, allowing the MPU direct access to the color palette RAM. MPU data is transferred into and out of the CacheDAC™ through the D0-D7 data pins. The read/write timing is controlled by the RD\* and WR\* inputs.

The RS0-RS3 select inputs specify which control register the MPU is accessing, as shown in Table 1. The 8-bit address register is used to address the color palette RAM, eliminating the requirement for external address multiplexers. D0 corresponds to ADDR0 and is the LSB.

## Hardware Reset Condition

On reset, Bt885 is configured for standard VGA compatibility as follows:

- 8 bits per pixel graphics, 1:1 MUX
- 6-bit DAC resolution
- Pixel mask register set to 0xFF
- Video modes disabled
- All control registers set for VGA compatibility
- Graphic pipelines are reset

## Writing Color Palette RAM Data

To write color data, the MPU writes the address register (RAM write mode) with the address of the color palette RAM location to be modified. The MPU performs three successive write cycles (8 bits each of red, green, and blue), using RS0-RS3 to select the color palette RAM. After the blue write cycle, the 3 bytes of color information are concatenated into a 24-bit word and written to the location specified by the address register. The address register then increments to the next location, which the MPU may modify by writing another sequence of red, green, and blue data. A block of color values in consecutive locations may be written to by writing the start address and performing continuous RGB write cycles until the entire block has been written. Refer to the Timing Waveforms section for further information.

## Reading Color Palette RAM Data

To read color palette RAM data, the MPU loads the address register (RAM read mode) with the address of the color palette RAM location to be read. The contents of the color palette RAM at the specified address are

copied into the red, green, or blue (RGB) registers and the address register is incremented to the next RAM location. The MPU performs three successive read cycles (8 bits each of red, green, and blue), using RS0-RS3 to select the color palette RAM. Following the blue read cycle, the contents of the color palette RAM at the address specified by the address register are copied into the RGB registers, and the address register increments again. A block of color values in consecutive locations may be read by writing the start address and performing continuous RGB read cycles until the entire block has been read.

RS3-RS0	Access	Addressed by MPU
0000	R/W	Address Register: Palette/Cursor RAM Write
0001	R/W	6/8-Bit Color Palette Data
0010	R/W	Pixel Mask Register
0011	R/W	Address Register: Palette/Cursor RAM Read
0100	R/W	Address Register: Cursor/Overscan Color Write
0101	R/W	Cursor Overscan and Color Data
0110	R/W	Command Register 0
0111	R/W	Address Register: Cursor/Overscan Color Read
1000	R/W	Command Register 1
1001	R/W	Command Register 2
1010	R/W	Extended Address Read/Write Register
1011	R/W	Cursor RAM Array Data
1100	R/W	Cursor x-Low Register
1101	R/W	Cursor x-High Register
1110	R/W	Cursor y-Low Register
1111	R/W	Cursor y-High Register

Table 1. Control Input Truth Table  
(RS3 = MSB, RS0 = LSB).

Circuit Description (continued)

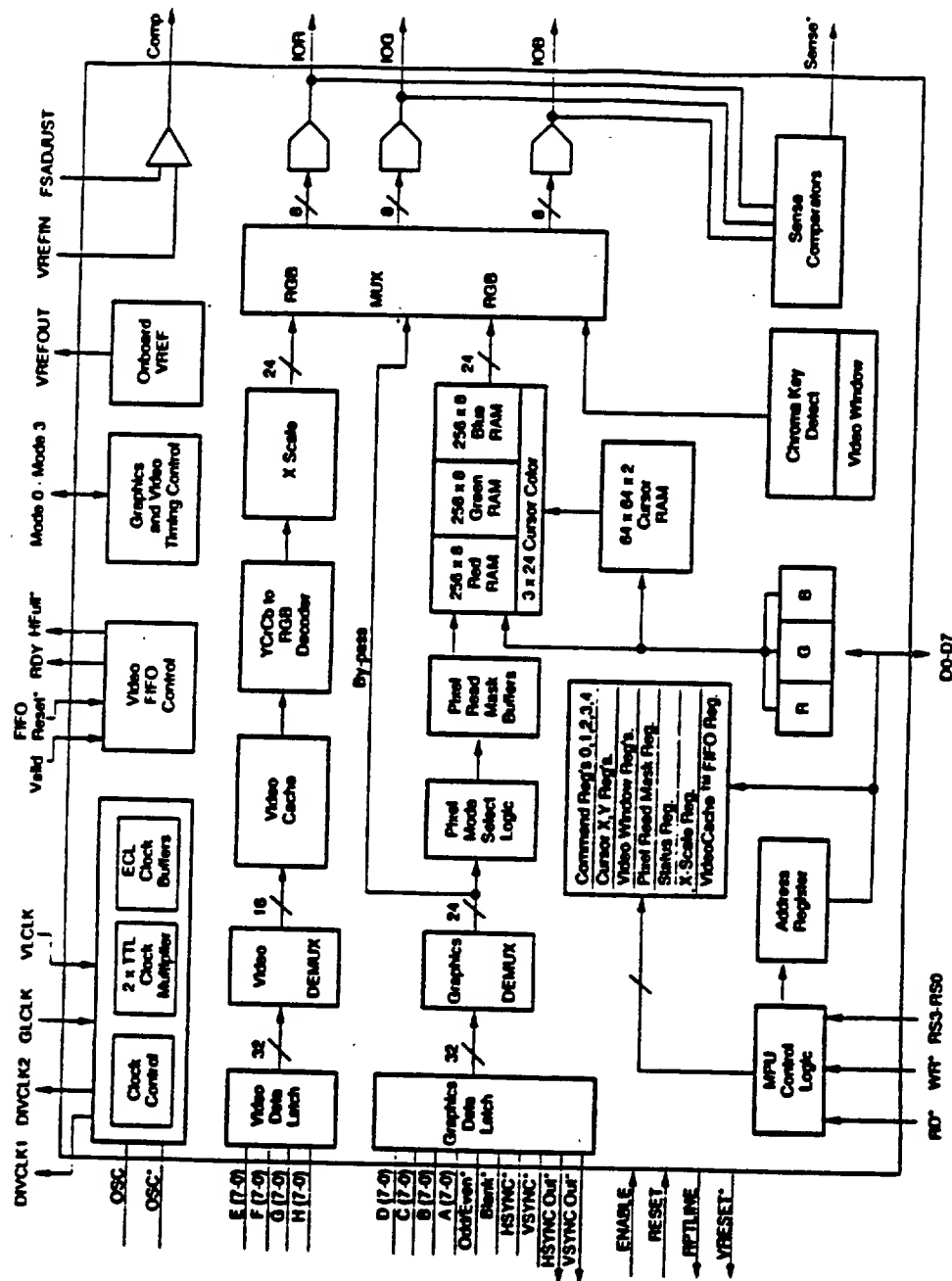


Figure 1. Bt885 Detailed Block Diagram.



## Circuit Description (continued)

**Writing Cursor and Overscan Color Data**

To write cursor or overscan color data, the MPU writes the address register (cursor color write mode) with the address of the cursor or overscan color location to be modified. The MPU performs three successive write cycles (8 bits each of red, green, and blue), using RS0–RS3 to select the cursor color registers. After the blue write cycle, the 3 bytes of red, green, and blue color information are concatenated into a 24-bit word and written to the cursor or overscan color location specified by the address register. The address register then increments to the next location, which the MPU may modify by writing another sequence of red, green, and blue data. A block of color values in consecutive locations may be written to by writing the start address and performing continuous RGB write cycles until the entire block has been written.

**Reading Cursor Color Data**

To read cursor color data, the MPU loads the address register (cursor color read mode) with the address of the cursor color location to be read. The contents of the cursor color register at the specified address are copied into the RGB registers, and the address register is incremented to the next cursor color location. The MPU performs three successive read cycles (8 bits each of red, green, and blue), using RS0–RS3 to select the cursor color registers. Following the blue read cycle, the contents of the cursor color location at the address specified by the address register are copied into the RGB registers, and the address register again increments. A block of color values in consecutive locations may be read by writing the start address and performing continuous R, G, B read cycles until the entire block has been read.

**Extended Register Mechanism**

An extended register set is used to accommodate all features of the Bt885. Since there are only four register select lines (and all 16 combinations have already been used), the extended registers must be accessed indirectly.

For example, Command Register 3 is accessed with the following sequence of operations:

1. Set RS3–RS0 = 0000, Address Register.
2. Write Address Register to 0x02.
3. Set RS3–RS0 = 1010 (Extended Address Register).
4. Read or Write Command Register 3.

**Writing Color Key Color Data**

To write the color key color data value, the MPU selects the color key data RGB register using the extended register. It then performs a write cycle setting RS3–RS0 to 1010 (Status Register). This process is repeated for each color component. The color key color register is only updated after the blue value is written.

**Reading Color Key Color Data**

To read the color key color data value, the MPU selects the color key data RGB register using the extended register mechanism, then performs a read cycle setting RS3–RS0 to 1010 (Status Register).

**Writing Color Key Mask Data**

To write the color key mask data value, the MPU selects the color key mask RGB register using the extended register mechanism. It then performs a write cycle setting RS3–RS0 to 1010 (Status Register). This process is repeated for each color component. The color key mask register is only updated after the blue value is written.

**Reading Color Key Mask Data**

To read the color key color mask value, the MPU selects the color key data RGB register using the extended register mechanism outlined below, then performs a read cycle setting RS3–RS0 to 1010 (Status Register).

**Additional Information**

When the color palette RAM is accessed, the address register resets to 0x00 following a blue read or write cycle to RAM location 0xFF.

The MPU interface operates asynchronously to the pixel clock. Data transfers between the color palette RAM and the color registers (R, G, and B in the functional block diagram) are synchronized by internal logic, and occur in the period between MPU accesses. To reduce noticeable sparkling on the CRT screen during MPU access to the color palette RAMs, internal logic maintains the previous output color data on the analog outputs while the transfer between RGB registers and lookup table RAMs occurs.

To keep track of the red, green, and blue read/write cycles, the address register has two additional bits (ADDRa and ADDRb) that count modulo three. They are reset to zero when the MPU writes to the address register and are not reset to zero when the MPU reads

## Circuit Description (continued)

the address register. The MPU does not have write access to these bits. The MPU may read the address register at any time without modifying its contents or the existing read/write mode. These bits can be read from SR1.

**Accessing the Cursor RAM Array**

The 64 x 64 x 2 cursor RAM is accessed in a planar format. Bits CR3\_1 and CR3\_0 in Command Register 3 become the load inputs to the 2 MSBs of a 10-bit address counter; therefore, these bits must be written in Command Register 3 before the lower 8 bits are written to the address counter through the MPU port. In the planar format, only nine address bits are used. The tenth bit determines which plane (0 or 1) data of the cursor RAM array is accessed. A single address presented to the cursor RAM accesses 8-bit locations in plane 0 or 1, depending on the state of address bit 9.

After each access in the planar format, the address increments. The MPU uses ADDR, a 10-bit binary address counter, to access the cursor RAM array. The address counter is the same 8-bit binary counter used for RGB autoincrementing with CR3\_1 and CR3\_0 as its extended MSBs. Any write to the address counter after cursor autoincrementing has been initiated resets the cursor autoincrementing logic until cursor RAM array has again been accessed. Cursor autoincrementing will then begin from the address written. A read from the address counter does not reset the cursor, autoincrementing logic. The color palette RAM and the cursor RAM share the same external address register, and MPU addressing for this and all other registers is determined by the external register select lines RS3–RS0 (see Table 2).

**6-Bit/8-Bit Operation**

The command bit CR0\_1 specifies whether the MPU is reading and writing 8 bits or 6 bits of color information each cycle. For an 8-bit operation, D0 is the LSB and D7 is the MSB of color data. For a 6-bit operation, color data is contained on the lower 6 bits of the data bus, with D0 as the LSB and D5 as the MSB of color data. When the MPU is writing color data, D6 and D7 are ignored. During color read cycles, D6 and D7 are a logical zero.

Accessing the cursor RAM array does not depend on the resolution of the DACs. When Bt885 is in the 6-bit mode, the 6-bit DAC values are left justified within an 8-bit field and the two LSBs are set to zero. Therefore, Bt885's full-scale output current will be about 1.5% lower than while it is in the 8-bit mode.

**Power-Down Mode**

The Bt885 incorporates a power-down capability, controlled by command bit CR0\_0. While command bit CR0\_0 is a logical zero, the Bt885 functions normally.

While command bit CR0\_0 is a logical one, the DACs, cursor circuitry, video FIFO, and power to the RAM are turned off. The RAM still retains the data. Also, the RAM may be read or written to by the MPU as long as the pixel clock is running. The RAM automatically powers up during MPU read/write cycles and shuts down when the MPU access is completed. The DACs output no current, and the four command registers may still be written to or read by the MPU. The output DACs require about one second to turn off (sleep mode) or turn on (normal), depending on the compensation capacitor used (see Table 11 in the Timing Waveforms section for further information). The DACs will be turned off during sleep mode only if a voltage reference (internal or external) is used.

When an external voltage reference is used, external circuitry should turn off the voltage reference (VREF = 0 V) to further reduce power consumption caused by biasing of portions of the internal voltage reference.

**Pixel Clock Selection**

OSC and OSC\* provide the source for the Bt885 internal pixel clock. Graphic pixel data is latched by GLCLK. Bit CR2\_4 selects whether the OSC or OSC\* pin is used. A clock doubler can be enabled on the selected input by setting CR3\_3 = 1. The OSC\* and OSC inputs can be used together as differential ECL inputs for the external clock by setting CR34 = 1. If a differential ECL input mode is used (CR34 = 1), then the state of CR2\_4 is ignored. The state of CR3\_3 must be 0.

It is also possible to internally route the DIVCLK2 output to the latches connected to GLCLK by setting CR3\_6 = 1. GLCLK will be ignored in this mode.

DIVCLK1 and DIVCLK2 are output on the basis of the OSC and OSC\* inputs as described unless they are disabled by setting CR3\_2 = 0 (DIVCLK1 disable) or CR3\_5 = 0 (DIVCLK2 disable). If the clock doubler is used (CR3\_3 = 1), then both the DIVCLK1 and DIVCLK2 dividers must be set to a value of 2 or greater. DIVCLK1 and DIVCLK2 are opposite phases.

## Circuit Description (continued)

CR3_1 (bit A9 of ADDR)	ADDR 0-7 (counts binary)	ADDR a,b (counts modulo 3)	RS3	RS2	RS1	RS0	Addressed by MPU
N/A	0x00-0xFF	00 01 10	0 0 0	0 0 0	0 0 0	1 1 1	Color Palette RAM (Red Component) Color Palette RAM (Green Component) Color Palette RAM (Blue Component)
N/A	0x00	00 01 10	0 0 0	1 1 1	0 0 0	1 1 1	Overscan Color (Red Component) Overscan Color (Green Component) Overscan Color (Blue Component)
N/A	0x01	00 01 10	0 0 0	1 1 1	0 0 0	1 1 1	Cursor Color 1 Red Component Cursor Color 1 Green Component Cursor Color 1 Blue Component
N/A	0x02	00 01 10	0 0 0	1 1 1	0 0 0	1 1 1	Cursor Color 2 Red Component Cursor Color 2 Green Component Cursor Color 2 Blue Component
N/A	0x03	00 01 10	0 0 0	1 1 1	0 0 0	1 1 1	Cursor Color 3 Red Component Cursor Color 3 Green Component Cursor Color 3 Blue Component
0 1	0x000-0x1FF 0x200-0x3FF	N/A N/A	1 1	0 0	1 1	1 1	Cursor RAM Array, Plane 0 Cursor RAM Array, Plane 1

Table 2. Address Register Operation and Autoincrementing.

## Frame Buffer Pixel Port Interface

There are 64 input pins P0-P7 (A-H) used to interface to the graphics and video frame buffer memories. The assignment of pins to input pixels is determined by the operation mode and multiplex rate.

## Video Port Clocking

Video data is synchronously clocked into Bt885 with the VLCLK input. VLCLK may be asynchronous from the pixel and/or graphics load clock, as an internal FIFO is used to synchronize video data to graphics pixel data.

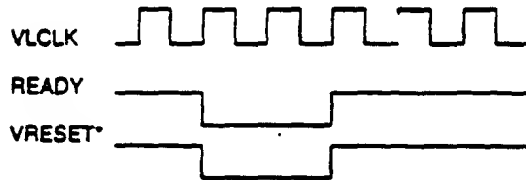
Three status signals are available to control the loading of video pixel data into Bt885: VALID, READY and HFULL. VALID is provided by the system to Bt885 and is asserted to indicate that valid

video data is being presented on the video pixel port. The READY signal is an output from Bt885 that indicates that it is accepting pixel data. For data to be accepted on any particular VLCLK rising edge, both the VALID and READY signals must be high through the clock edge. The HFULL signal is used to keep check on how full the video cache is and helps to prevent overloading the internal video FIFO.

The system must load video data into Bt885 prior to the time that it is to be used. In systems where there is a one-to-one relationship between video pixels and graphics pixels in the frame buffer and this data is delivered simultaneously, the FIFO operation can be ignored and VALID would be tied to the pixel blanking signal from the graphics subsystem (BLANK\*). In this mode, the FIFO would never be filled and, therefore, READY may be ignored.

## Circuit Description (continued)

The internal video data FIFO is reset to an empty state on each detected vertical blank period. The system can immediately begin loading data into the video port regardless of the video window's position on the screen. With the VRESET\* signal (see Pin Descriptions section) the video data will know the start of each graphics frame. See diagram below:



If at any time the video FIFO is empty when video data is required, Status Register 2 bit SR27 will be set to one. The underflow bit will remain set until Status Register 2 is written, then SR27 will be cleared.

For proper operation of the video pipeline reset, VLCLK must be a free-running clock.

## VideoCache™ FIFO Operation

The Bt885 provides a FIFO buffer for video pixels to allow for asynchronous video and graphics operation, and to ease system design requirements. Use of the VideoCache™ FIFO features is entirely optional and not necessary for synchronous designs.

## Loading VideoCache™ FIFO

The VideoCache™ FIFO accepts a group of data (the exact number is given by the current video mode) when the following conditions are met on any single rising edge of VLCLK:

1. The FIFO is ready to accept data (i.e., it is not full). This is determined by the state of the READY signal.
2. The system is presenting data, indicating this to the CacheDAC™ by asserting the VALID signal with the data.

## Unloading VideoCache™ FIFO

Bt885 will unload the VideoCache™ FIFO dependant on the setting of bit CR4\_1. If CR4\_1 = 1, the video will only be unloaded while Bt885 is scanning through the video window. If CR4\_1 = 0, then video will always be unloaded during active graphics time. The unloading process is independent of color keying.

## HFULL

This signal is asserted when the VideoCache™ FIFO gets more than half full.

## VideoCache™ Reset

There are four ways that the VideoCache FIFO gets reset:

**FIFO Reset Pin.** This is an external hardware FIFO RESET method for resetting the Bt885 Video FIFO. This pin must be held low for at least two VLCLKs with PDXEL CLOCK running.

**CR4\_7.** This is a software RESET method for resetting the Bt885 VideoCache FIFO. A logical one written to this bit resets the VideoCache FIFO after four VLCLKs. A logical zero will put the FIFO back to normal operation.

**An underflow occurred.** The SR2\_7 status bit says a VIDEO FIFO underflow occurred when a logical one is read. The VideoCache FIFO is automatically reset when this happens. A MPU write cycle to Status Register 2 will clear SR27.

**Vertical Retrace Interval.** An automatic VIDEO FIFO reset also occurs during the vertical retrace interval. When the ENABLE line is low for 2048 clock cycles an internal FIFO RESET sequence is initiated.

READY and VRESET\* will be = 0 during the FIFO reset period. READY will become active within one PDXEL CLOCK period after VRESET\* goes high.

## General Purpose Signals

## DIVCLK1 / DIVCLK2

These signals provide programmable free-running clocks based on the internal pixel clock. They can be used to generate external pixel load clocks, such as VLCLK or GLCLK. A gated clock may be generated from DIVCLK1 by using another general purpose signal, SEN, described below.

## SEN

This signal is used to provide a gating control for DIVCLK1. SEN can be programmed to start relative to the falling edge of internally detected vertical blank (see cursor operation) in units of scanlines and relative to the falling edge of C/HSYNC\* in DIVCLK1 cycles using the serial clock enable start (horizontal and vertical) registers. Duration is set in units of scanlines for the vertical direction and in DIVCLK1 cycles for the horizontal direction (relative to the beginning of SEN) using the serial clock enable duration (horizontal and vertical) registers. This signal is guaranteed to transition only during DIVCLK1 low time.

## Circuit Description (continued)

This signal may be used to control a VRAM shift clock which runs during non-blanking time. When an appropriate delay is programmed from the leading edge of  $\overline{CHSYNC^*}$ , the serial data can be properly positioned before the trailing edge of  $\overline{BLANK^*}$ . The SEN duration register then stops the serial clock to allow the system to perform VRAM row data transfer. Because  $\overline{CHSYNC^*}$  is sampled with the internal pixel clock, there may be an additional pixel clock delay between  $\overline{CHSYNC^*}$  falling and the SEN rising.

## MODE0

This is a general purpose, TTL compatible, registered input/output which is set using CR4\_5. Selection of input or output is made using CR3\_7.

This pin is user-definable and could be used to interface between hardware and software. For example, MODE0 could detect the existence of a video card. Software would detect this by reading CR4\_5.

## Video Window Operation

The XSTART register indicates the starting X position on the screen for the video window relative to the ENABLE pin (Figure 2). A value of zero indicates that

the video window begins with the first (leftmost) pixel of each horizontal scan line. The YSTART register indicates the starting Y position on the screen for the video window. A value of zero indicates that the video window begins on the first scan line of each frame. The XWIDTH register indicates the number of pixels per scan line within the video window. A value of zero indicates that there are no pixels in the video window. The YHEIGHT register indicates the number of scan lines within the video window. A value of zero indicates that there are no scan lines in the video window.

All four values, XSTART, XWIDTH, YSTART, and YHEIGHT should be written sequentially. Internal video window coordinates are loaded during the next detected vertical blanking interval after the YHEIGHT register is written.

## Video Scaling Operation

The Bt885 supports video upscaling in the horizontal direction. Horizontally, a combination of coarse pixel interpolation and pixel-accurate replication may be applied. Downscaling of the source image, both horizontally and vertically, must be performed outside the Bt885.

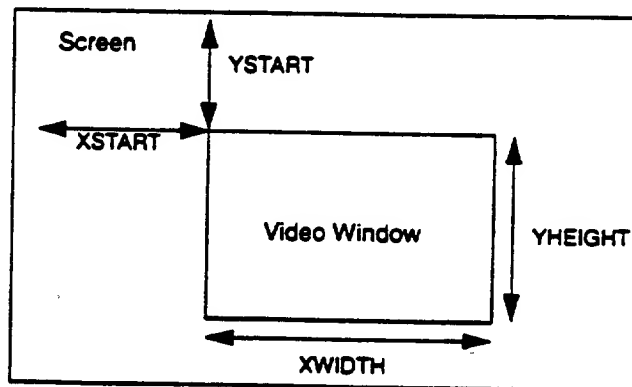


Figure 2. Video Window Registers.

## Circuit Description (continued)

## Horizontal Scaling

Horizontal upscaling may be accomplished by using a combination of two methods: pixel replication and pixel interpolation.

Pixel replication is accomplished by using the output of an overflowing 12-bit accumulator to either clock a value out of the VideoCache™ FIFO to the DACs or to hold the current DAC value.

At the start of each scan line, the accumulator is initialized to the value stored in the XSCALEINTT register. On each pixel, the value stored in the XSCALEINC register is added to the accumulator.

If the addition results in a carry, a pixel is clocked out of the VideoCache™ FIFO to the DACs. If no carry occurs, the previous DAC value is held. This style of scaling is known as a Digital Differential Algorithm (DDA).

To accomplish scaling, the system supplying the Bt885 with video pixel must precalculate the DDA constants required for the desired scale factor and load the values into the two 12-bit X-scaling registers, XSCALEINTT and XSCALEINC, as follows:

$$\text{XSCALEINC} = [(\text{Source Video Width} \times 0x1000) + 0x0800 - \text{XSCALEINTT}] / \text{Destination Video Width}$$

$0 \leq \text{XSCALEINTT} \leq 0x0FFF$ . XSCALEINTT can be used to set the replication phase of the DDA in more advanced applications.

Pixel interpolation is available when using certain YCrCb video modes. When used, it can interpolate the data to 2x or 4x the source horizontal pixel count. The table below shows source data formats, video mode selected (CR1), and the resulting interpolation factor achieved.

Source Video Format	Video Mode Selected	Interpolation Factor Achieved
YCrCb 4:2:2	YCrCb 4:2:2	1:1
	YCrCb 2:1:1	2:1
	YCrCb 1:1:1	4:1
YCrCb 2:1:1	YCrCb 2:1:1	1:1
	YCrCb 1:1:1	2:1
YCrCb 1:1:1	YCrCb 1:1:1	1:1

**Example:** To fill a window which is 636 pixels wide with a source of 320 pixels of YCrCb 4:2:2 data loaded

in 1:1 mux mode, one should select the YCrCb 2:1:1 video mode (CR1\_7-CR1\_4 = 5) and set XSCALEINC to 0x0FFF for no replication. If the window were slightly larger, say 700 pixels wide, one should select the YCrCb 2:1:1 video mode and use the pixel replicator to stretch the 636 new pixels into 700 pixels (XSCALEINC = 0x0E89, XSCALEINTT = 0x0800).

## Color Key Operation

Selection between the video and graphics pixel data may be based on a specified range of graphic pixel values. A "color key set" may be defined which specifies one or more graphic pixel values that allow video pixels to be shown.

To define the color key set, three color key registers and three color mask registers are used. A graphic pixel value is bitwise XORed with the color key and the result is NANDed with the color mas. If the result is one, the corresponding video pixel is displayed in its place.

When a graphic pixel value falls within the color key set, the corresponding video pixel is displayed rather than the graphic pixel. Color key detection may occur either before the palette lookup or after the palette lookup. In 16- and 24-bit pixel modes, if palette bypass is enabled, selecting matching after the palette matches based on the actual values that would be applied to the actual values that would be applied to the DACs.

When matching after the palette, bit CR4\_2 of Command Register 4 should be set to 1, and the color key registers and color mask registers represent 24-bit RGB values each. The registers are ordered with red at the lowest address, then green and blue.

When matching before the palette, bit CR4\_2 of Register 4 should be set to zero. The color key registers and color mask registers represent unmultiplexed graphic pixel values, with the red register as the least significant byte, then green and blue. Only the bits needed to represent the pixel are used. For example, and 8-bit pixel color key and mask use only the red gisters, 16-bit pixels use only the red and green registers.

Pixel selection occurs only within the current video window boundaries, and only when bit CR4\_6 of Command Register 4 is set to 0 to allows color key detection. When CR4\_6 is set to 1, all pixels within the video window ill display the video pixels, regardless of color mask and key register values.

The hardware cursor always has display priority over color key selection.

## Circuit Description (continued)

## Example 1

Match a specific 8-bit pseudo-color palette position (value 0 x FE).

CR4\_2 = 0 (matching before palette)

CR4\_6 = 0 (allows color keying)

Color Mask: (B) 0xXX (G) 0xXX (R) 0xFF

Color Key: (B) 0xXX (G) 0xXX (R) 0xFE

## Example 2

Match a range of blue values between 0xC0 and 0xC7.

CR4\_2 = 0 (matching after palette)

CR4\_6 = 0 (allows color keying)

Color Mask: (B) 0xF8 (G) 0x00 (R) 0x00

Color Key: (B) 0xC0 (G) 0x00 (R) 0x00

## Example 3

Use bit 15 in a TARGA 15-bit true-color mode to perform color key.

CR4\_3 = 0 (matching before palette)

CR4\_6 = 0 (allows color keying)

Color Mask: (B) 0xXX (G) 0x80 (R) 0x00

Color Key: (B) 0xXX (G) 0x80 (R) 0x00

*Note: To set the color key or color mask, all three indexes must be written, even if all three indexes are not used.*

## YCrCb-to-RGB Matrix

The matrix converts the YCrCb video data to 24 bits of RGB data (8 bits each).

The YCrCb-to-RGB conversion is compliant with CCIR Recommendation 601-1 as follows:

$$R = 1.164(Y - 16) + 1.596(Cr - 128)$$

$$G = 1.164(Y - 16) - 0.813(Cr - 128) - 0.391(Cb - 128)$$

$$B = 1.164(Y - 16) + 2.018(Cb - 128)$$

## Modes of Operation—Graphics

## 4-Bits/Pixel Operation (8:1 MUX)

The 32 input bits are multiplexed 8:1 and configured for 4 bits/pixel. There are eight independent 4-bit pixel ports, P7:4 (A-D) and P3:0 (A-D). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every eight pixel clock cycles. The 4 bits from each port will select one of 16 locations in the palette (see Table 14 in the Internal Registers section).

## 8-Bits/Pixel Operation (4:1 MUX)

The 32 input bits are multiplexed 4:1 and configured for 8 bits/pixel. There are four independent 8-bit pixel ports, (A-D). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every four pixel clock cycles. The 8 bits from each port will select 1 of 256 locations in the palette (see Table 14 in the Internal Registers section).

## 8-Bits/Pixel Operation (2:1 MUX)

The 16 input bits are multiplexed 2:1 and configured for 8 bits/pixel. There are two independent 8-bit pixel ports, (A-B). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every two pixel clock cycles. The 8 bits from each port will select 1 of 256 locations in the palette (see Table 14 in the Internal Registers section).

## 8-Bits/Pixel Operation (1:1 MUX)

The 8 input bits are multiplexed 1:1 and configured for 8 bits/pixel. There is one 8-bit pixel port, (A). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every pixel clock cycle. The 8 bits will select 1 of 256 locations in the palette (see Table 14 in the Internal Registers section).

## 16-Bits/Pixel Operation (2:1 MUX)

The 32 input bits are multiplexed 2:1 and configured for 16 bits/pixel. There are two independent 16-bit pixel ports, (A-B) and (C-D). The bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every two pixel clock cycles. The pixel bits multiplexed in this mode are from the same ports of RGB color formats of 5:5:5 or 5:6:5. P7D and P7B are ignored internally when the 5:5:5 color format is selected (see Table 14 in the Internal Registers section).

Bit CR2\_4 in Command Register 2 can be programmed to enable or disable true-color palette bypass. When the bypass mode is selected, the pixel data bypasses the palette as well as the pixel mask, and is transferred to the proper MSBs of the respective DACs, the remaining LSBs are set to zeros. When the bypass mode is not selected, the pixel data indexes the palette, and color information is passed to the respective DACs. Bit CR2\_2 in Command Register 2 determines whether palette addressing is sparse or contiguous. For sparse palette addressing, each independent color component of pixel data is mapped to the MSBs of the respective palette address; the LSBs are set to zero. For contiguous palette addressing, each independent color component of the pixel data is mapped to the LSBs of the respective palette address; the MSBs are set to zero. The color palette values indexed, for either sparse or contiguous addressing, are

## Circuit Description (continued)

transferred to the DACs. When 5:5:5 or 5:6:5 color format is selected, the display can contain 32 K or 64 K simultaneous colors. The DACs can be configured for 6 or 8 bits of resolution in this mode. If 5:5:5 color format is selected, the MSB may be used for color key operation (see Table 3 and Table 4).

## 16-Bits/Pixel Operation (1:1 MUX)

The 16-bit pixel port (A-B) is latched on the rising edge of GLCLK and is multiplexed 1:1. One rising edge of GLCLK should occur every pixel clock cycle.

Bit CR2\_5 in Command Register 2 can be programmed to enable or disable true-color palette bypass. When the bypass mode is selected, the pixel data bypasses the palette as well as the pixel mask, and is transferred to the proper MSBs of the respective DACs. When the bypass mode is not selected, the pixel data indexes the proper locations in the palette, and the correct color information is passed to the respective DACs. Bit CR2\_2 in Command Register 2 determines whether palette addressing is sparse or contiguous. For sparse palette addressing, each independent color component of pixel data is mapped to the most significant bits of the respective palette address; the LSBs are set to zero. For contiguous palette addressing, each independent color component of the pixel data is mapped to the LSBs of the respective palette address; the MSBs are set to zero. The color palette values indexed, for either sparse or contiguous addressing, are transferred to the DACs. When 5:5:5 or 5:6:5 color format is selected, the display can contain 32 K or 64 K simultaneous colors. The DACs can be configured for 6 or 8 bits of resolution in this mode (see Table 3 and Table 4).

If 5:5:5 color format is selected, the MSB may be used for color key operation.

For graphics pixel index masking, see Table 5.

## 24-Bits/Pixel Operation (1:1 MUX)

When 24 bits/pixel in 1:1 MUX mode is selected, there is one 24-bit pixel port. (A-C). The pixel bits are latched on the rising edge of GLCLK and multiplexed 1:1. One rising edge of GLCLK should occur every pixel clock cycle. The RGB color format in this mode is 8:8:8.

Bit CR2\_5 in Command Register 2 can be programmed to enable or disable true-color palette bypass. When the bypass mode is selected, the pixel data bypasses the palette as well as the pixel mask, and is transferred to the proper MSBs of the respective DACs. When the bypass mode is not selected, the pixel data indexes the proper locations in the palette, and the independent RGB color values are passed to the respective DACs (see Table 6a thru Table 6c). When 8:8:8 color format is selected, the display can contain 16.8 million simultaneous colors. The DACs should be configured for 8 bits of resolution in this mode (CR25 = 1, CR0\_1 = 1). CR4\_1 and CR4\_0 can be used to alter the pixel read order to BRG or BGR.

## Pixel Read Mask Register

The pixel data can be masked before being transferred to the color palette with the 8-bit pixel mask register. The pixel data is bit-wise logically ANDed with the contents of the pixel read mask register. The result is used to address the color palette RAM. The addressed location provides 24 bits of color information to the three D/A converters. Pixel masking is enabled for all modes of operation except when the true-color bypass is enabled. The pixel mask register is initialized to logical ones at reset (see Table 16, Register Values on Reset in the Internal Register section).



## Circuit Description (continued)

Bit	MSB															LSB
Format	X	R	R	R	R	R	G	G	G	G	G	B	B	B	B	B
Port 1	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A
Port 2	P7D	P6D	P5D	P4D	P3D	P2D	P1D	P0D	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C

Note: X bit may be used for color key before the palette.

Table 3. 5:5:5 RGB Graphics Color Format for Both 2:1 and 1:1 Multiplexing Modes.

Bit	MSB															LSB
Format	R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B
Port 1	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A
Port 2	P7D	P6D	P5D	P4D	P3D	P2D	P1D	P0D	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C

Table 4. 5:6:5 RGB Graphics Color Format for Both 2:1 and 1:1 Multiplexing Modes.

	MSB							LSB	X=Map to Zero
Pixel Mask Register	7	6	5	4	3	2	1	0	Register Bits
4 Bits/Pixel	x	x	x	x	3	2	1	0	Palette Index
8 Bits/Pixel	7	6	5	4	3	2	1	0	Palette Index
16 Bits/Pixel 5:5:5 Format SPARSE	7 7 7	6 6 6	5 5 5	4 4 4	3 3 3	x x x	x x x	x x x	Red Palette Index Green Palette Index Blue Palette Index
16 Bits/Pixel 5:5:5 Format CONTIGUOUS	x x x	x x x	x x x	4 4 4	3 3 3	2 2 2	1 1 1	0 0 0	Red Palette Index Green Palette Index Blue Palette Index
16 Bits/Pixel 5:6:5 Format SPARSE	7 7 7	6 6 6	5 5 5	4 4 4	3 3 3	x x x	x x x	x x x	Red Palette Index Green Palette Index Blue Palette Index
16 Bits/Pixel 5:6:5 Format CONTIGUOUS	x x x	x x x	x x x	4 4 4	3 3 3	2 2 2	1 1 1	0 0 0	Red Palette Index Green Palette Index Blue Palette Index
24 Bits/Pixel 8:8:8 Format	7 7 7	6 6 6	5 5 5	4 4 4	3 3 3	2 2 2	1 1 1	0 0 0	Red Palette Index Green Palette Index Blue Palette Index

Note: x means final DAC bit will be 0

Table 5. Graphics Pixel Index Masking.

Circuit Description (continued)

Bit	MSB																							LSB
Format	R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B
Port 1	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A

Table 6a. 24-Bits/Pixel Graphics RGB Color Format (CR4\_1, 40 = 00) for 1:1 MUX Modes.

Bit	MSB																							LSB
Format	B	B	B	B	B	B	B	B	R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G
Port 1	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A

Table 6b. 24-Bits/Pixel Graphics BRG Color Format (CR4\_1, 40 = 01).

Bit	MSB																							LSB
Format	B	B	B	B	B	B	B	B	G	G	G	G	G	G	G	G	R	R	R	R	R	R	R	R
Port 1	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A

Table 6c. 24-Bits/Pixel Graphics BRG Color Format (CR4\_1, 40 = 10).

**Circuit Description** (continued)**Modes of Operation—Video**

The pixel ordering and YCrCb-to-RGB conversions are shown in Table 7 through Table 9. The video pixel port configuration is shown in Table 14 in the Internal Registers section. The following describes video operation modes.

**YCrCb 1:0.5:0.5 Operation (4 Bytes/8 Pixels)**

The 32 input bits are configured for YCrCb 1:0.5:0.5. There are four independent 8-bit pixel ports, (E–H). Each group of 4 bytes results in 8 output pixels. The pixel bits are latched on the rising edge of VLCLK.

YCrCb 1:0.5:0.5 Video Color Format (4 Bytes/8 Pixels)																
Bit	MSB															LSB
Format	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

**YCrCb 1:0.5:0.5 Operation (2 Bytes/4 Pixels)**

The 16 input bits are configured for YCrCb 1:0.5:0.5. There are two independent 8-bit pixel ports, (G–H). Each group of 2 bytes results in 4 output pixels. The pixel bits are latched on the rising edge of VLCLK.

YCrCb 1:0.5:0.5 Video Color Format (4 Bytes/4 Pixels)																
Bit	MSB															LSB
Format	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

**YCrCb 1:0.5:0.5 Operation (1 Byte/2 Pixels)**

The 8 input bits are configured for YCrCb 1:0.5:0.5. There is one 8-bit pixel port, (H). Each byte loaded results in 2 output pixels. The pixel bits are latched on the rising edge of VLCLK.

CrCb 1:0.5:0.5 Video Color Format (4 Bytes/2 Pixels)																
Bit	MSB															LSB
Format	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	X	X	X	X	X	X	X	X
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

## Circuit Description (continued)

## YYCrCb 2:1:1 Operation (4 Bytes/4 Pixels)

The 32 input bits are configured for YYCrCb 2:1:1. There are four independent 8-bit pixel ports. (E-H). The pixel bits are latched on the rising edge of VLCLK.

YYCrCb 1:0.5:0.5 Video Color Format (4 Bytes/4 Pixels)																
Bit	MSB															LSB
Format	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

## YCrCb 2:1:1 Operation (2 Bytes/2 Pixels)

The 16 input bits are configured for YCrCb 2:1:1. There are two independent 8-bit pixel ports. (G-H). The pixel bits are latched on the rising edge of VLCLK.

YCrCb 12:1:1 Video Color Format (2 Bytes/2 Pixels)																
Bit	MSB															LSB
Format	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

## YCrCb 2:1:1 Operation (1 Byte/1 Pixel)

The 8 input bits are configured for YCrCb 2:1:1. There is one 8-bit pixel port. (H). Each byte loads results in two output pixels. The pixel bits are latched on the rising edge of VLCLK.

YCrCb 2:1:1 Video Color Format (1 Byte/1 Pixel)																
Bit	MSB															LSB
Format	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	X	X	X	X	X	X	X	X
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

**Circuit Description** *(continued)***YCrCb 4:2:2 Operation (4 Bytes/2 Pixels)**

The 32 input bits are configured for YCrCb 4:2:2. There are four independent 8-bit pixel ports. (F-E) and (H-G). The bits are latched on the rising edge of VLCLK.

YCrCb 4:2:2 Video Color Format (4 Bytes/2 Pixels)																
Bit	MSB															LSB
Format	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7L	P6E	P5E	P4E	P3E	P2E	P1E	P0E

**YCrCb 4:2:2 Operation (2 Bytes/1 Pixel)**

The 16 input bits are configured for YCrCb 4:2:2. There are two independent 8-bit pixel ports. (G-H). The input bits are latched on the rising edge of VLCLK.

YCrCb 4:2:2 Video Color Format (2 Bytes/1 Pixel)																
Bit	MSB															LSB
Format	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

**16-Bits/Pixel 5:5:5 Operation (2:1 MUX)**

The 32 input bits are configured for 16 bits/pixel. There are two independent 16-bit pixel ports. (E-F) and (G-H). The bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:5:5. The most significant bit is not used.

**16-Bits/Pixel 5:5:5 Operation (1:1 MUX)**

The 16 input bits are configured for 16 bits/pixel. There is one 16-bit pixel port. (G-H). The input bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:5:5. The most significant bit is not used.

5:5:5 RGB Video Color Format for Both 2:1 and 1:1 Multiplexing Modes																
Bit	MSB															LSB
Format	A	R	R	R	R	R	G	G	G	G	G	B	B	B	B	B
Port 1	P7C	P6G	P5G	P4G	P3G	P2G	P1G	P0G	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H
Port 2	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F

## Circuit Description (continued)

## 16-Bits/Pixel 5:6:5 Operation (2:1 MUX)

The 32 input bits are configured for 16 bits/pixel. There are two independent 16-bit pixel ports, (E-F) and (G-H). The bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:6:5.

## 16-Bits/Pixel 5:6:5 Operation (1:1 MUX)

The 16 input bits are configured for 16 bits/pixel. There is one 16-bit pixel port, (G-H). The input bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:6:5.

5:6:5 RGB Video Color Format for Both 2:1 and 1:1 Multiplexing Modes																
BR	MSB															LSB
Format	R	R	R	R	R	R	G	G	G	G	G	B	B	B	B	B
Port 1	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H
Port 2	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F

## 24-Bits/Pixel Operation (1:1 MUX)

The 24 input bits are configured for 24 bits/pixel. There is one 24-bit pixel port, (F-H). The bits are latched on the rising edge of VLCLK. The RGB or BGR color format in this mode is 8:8:8. The color format is controlled by bit CR4\_4 in Command Register 4.

24-Bit RGB Video Color Format (CR4_4=0) for 1:1 Multiplexing Modes																
BR	MSB															LSB
Format	R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G
Port 1	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	B	B	B	B	B	B	B	B								
Port 2	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F

4-Bit RGB Video Color Format (CR4_4=1) for 1:1 Multiplexing Modes																
BR	MSB															LSB
Format	B	B	B	B	B	B	B	B	G	G	G	G	G	G	G	G
Port 1	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	R	R	R	R	R	R	R	R								
Port 2	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H								

## Circuit Description (continued)

CCIR601 1:0.5:0.5

CCIR656 Component Ordering

Color Space: YCrCb

Subsampling: 1:0.5:0.5

Address (8N+)	0	1	2	3	4	5	6	7
Value	Cb0	Y0	Cr0	Y4	Cb8	Y8	Cr8	Y12

Pixel	0	1	2	3	4	5	6	7
Y	Y0	$\frac{3Y0 + Y4}{4}$	$\frac{Y0 + Y4}{2}$	$\frac{Y0 - 3Y4}{4}$	Y4	$\frac{3Y4 + Y8}{4}$	$\frac{Y4 + Y8}{2}$	$\frac{Y4 + 3Y8}{4}$
Cr	Cr0	Cr0	$\frac{3Cr0 + Cr8}{4}$	$\frac{3Cr0 + Cr8}{4}$	$\frac{Cr0 + Cr8}{2}$	$\frac{Cr0 + Cr8}{2}$	$\frac{Cr0 + 3Cr8}{4}$	$\frac{Cr0 + 3Cr8}{4}$
Cb	Cb0	Cb0	$\frac{3Cb0 + Cb8}{4}$	$\frac{3Cb0 + Cb8}{4}$	$\frac{Cb0 + Cb8}{2}$	$\frac{Cb0 + Cb8}{2}$	$\frac{Cb0 + 3Cb8}{4}$	$\frac{Cb0 + 3Cb8}{4}$

Table 7. CCIR601 1:0.5:0.5 Video Format.

CCIR601 2:1:1

CCIR656 Component Ordering

Color Space: YCrCb

Address (8N+)	0	1	2	3	4	5	6	7
Value	Cb0	Y0	Cr0	Y2	Cb4	Y4	Cr4	Y6

Pixel	0	1	2	3	4	5	6	7
Y	Y0	$\frac{Y0 + Y2}{2}$	Y2	$\frac{Y2 + Y4}{2}$	Y4	$\frac{Y4 + Y6}{2}$	Y6	$\frac{Y6 + Y8}{2}$
Cr	Cr0	$\frac{3Cr0 + Cr4}{4}$	$\frac{Cr0 + Cr4}{2}$	$\frac{Cr0 + 3Cr4}{4}$	Cr4	$\frac{3Cr4 + Cr8}{4}$	$\frac{Cr4 + Cr8}{2}$	$\frac{Cr4 + 3Cr8}{4}$
Cb	Cb0	$\frac{3Cb0 + Cb4}{4}$	$\frac{Cb0 + Cb4}{2}$	$\frac{Cb0 + 3Cb4}{4}$	Cb4	$\frac{3Cb4 + Cb8}{4}$	$\frac{Cb4 + Cb8}{2}$	$\frac{Cb4 + 3Cb8}{4}$

Table 8. CCIR601 2:1:1 Video Format.

**Circuit Description** *(continued)*

CCIR601 4:2:2

CCIR656 Component Ordering

Color Space: YCrCb

Address (8N+)	0	1	2	3	4	5	6	7
Value	Cb0	Y0	Cr0	Y1	Cb2	Y2	Cr2	Y3

Pixel	0	1	2	3
Y	Y0	Y1	Y2	Y3
Cr	Cr0	$\frac{Cr0 + Cr2}{2}$	Cr2	$\frac{Cr2 + Cr4}{2}$
Cb	Cb0	$\frac{Cb0 + Cb2}{2}$	Cb2	$\frac{Cb2 + Cb4}{2}$

**Table 9. CCIR601 4:2:2 Video Format.****DAC Values In 16-Bits/Pixel Video Modes**

In order to achieve 8-bit full-scale DAC output in the 5:5:5 16-bits/pixel video modes, each 5-bit value will be used as the 5 MSBs of the 8-bit DAC value and the 3 MSBs of the 5-bit pixel value will be duplicated in the low order 3 bits before the pixel value is passed to the

DACs. Similarly, in 5:6:5 modes, when processing the 6-bit green component, the 6-bit value will be used as the 6 MSBs of the 8-bit DAC value and the 2 MSBs of the 6-bit pixel value will be duplicated in the low order 2 bits before the pixel value is passed to the DACs.



## Circuit Description (continued)

**Cursor Operation**

The Bt885 has an on-chip, three-color,  $64 \times 64 \times 2$  pixel user-definable cursor. This cursor works with both interlaced and noninterlaced systems. The cursor always has display priority over both video and graphics pixels.

The pattern for the cursor is provided by the cursor RAM, which may be accessed by the MPU at any time. The cursor is positioned through the cursor position register (Xp,Yp) (see Figure 3). A (0,0) written to the cursor position registers will place the cursor completely offscreen. A (1,1) written to the cursor position registers will place the lower right pixel of the cursor on the upper left corner of the screen. Only one cursor pattern per frame is displayed at the location specified for both interlaced and noninterlaced display formats, regardless of the number of updates to (Xp,Yp). The cursor's vertical or horizontal location is not affected during any frame displayed.

There are no restrictions on updating (Xp,Yp) other than both cursor position registers must be written when the cursor location is updated. Internal x and y position registers are loaded after the upper byte of Yp has been written to ensure one cursor pattern per frame at the correct location. The cursor pattern is displayed

at the last cursor location written. Cursor positioning is relative to ENABLE. The cursor position is not dependent upon BLANK<sup>\*</sup> (see Figure 3). The cursor Xp position is relative to the first rising edge of GLCLK when ENABLE is sampled at logical one. The cursor Yp position is relative to the first rising edge of GLCLK when ENABLE is sampled at logical one after the ENABLE vertical blanking interval has been determined (see Figure 3). If an ENABLE transition from logical zero to logical one (as determined by GLCLK) does not occur within 2048 internal pixel clocks, ENABLE is in vertical blanking.

For proper cursor operation, selection of interlaced or noninterlaced cursor display must be set using bit CR2\_3 in Command Register 2.

Figure 4 is a visual explanation of planar pixel format and cursor RAM array pixel mapping.

While the cursor may be disabled by setting bits CR2\_0-21 of Command Register 2 to zero, this practice is not recommended. The recommended method for disabling the cursor is to move it entirely offscreen by setting the cursor X and Y location registers to (0,0).

Table 10 gives the pseudo code for Bt885 to check for monitor connection.

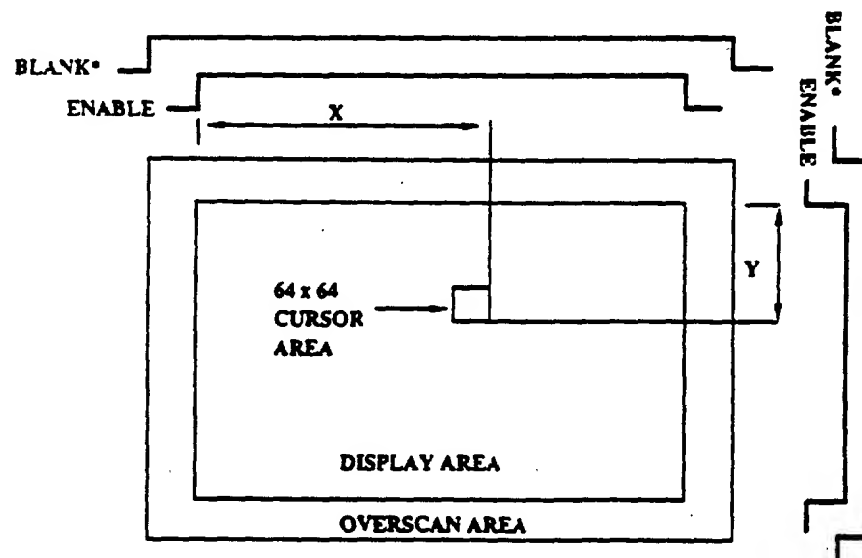


Figure 3. Cursor Positioning.

## Circuit Description (continued)

This is Pseudo Code for Bt885 to check for monitor connection. Problem: Verify if an RGB or single input monitor is connected to the RAMDAC Analog Outputs	
Program Monitor?	(*Verify Monitor Connection*)
Reset Bt885	(*Toggle RESET* of Bt885*)
Set C/HSYNC* = low	(*Disable SYNC Current*)
Set BLANK = high	(*Enable RamDAC Outputs*)
Set Pixel Mask = \$00	(*Disable external Pixel Input*)
Set RGB LUT Loc.0 = \$18	(*10.3 mV x 24 = 247 mV*)
Read Status Register	(*Check on State of SENSE**)
If SR1-3 = 1	(*Check for RGB Monitor Connection*)
RGB Monitor	
ELSE	
Single Monitor?	
Address LUT Loc. \$00	(*Set Address Register to Program Lut Loc. 0*)
Set Red DAC Output = \$00	(*Set Red DAC to Output 0 mV*)
Set Grn DAC Output = \$18	(*Set Green DAC to Output 247 mV*)
Set Blue DAC Output = \$00	(*Set Blue DAC to Output 0 mV*)
Read Status Register	(*Check for Single Input Monitor on Grn.*)
If SR1-3 = 1	
Single Input Monitor	
ELSE	
NO Monitor Sensed	
End	

Table 10. Pseudo Code for Bt885 Monitor Connection.

Figure 5 and Figure 6, and Table 12 and Table 13 detail how the C/HSYNC\* and BLANK\* inputs modify the output levels.

### Highlight Logic

The highlight logic is enabled in cursor mode 2 when plane 1 and plane 0 data are logical ones (see Table 11). When the highlight logic is enabled, it ensures that the graphics pixel highlighted has a unique color. This is because the highlight logic bit-wise complements the 24 (18)-bit graphics palette or bypass data supplied to the DACs.

## Video Generation

The C/HSYNC\* and BLANK\* inputs are latched on the rising edge of GLCLK to maintain synchronization with the pixel data.

Pipelined C/HSYNC\* and VSYNC\* are output on the C/HSYNC\* OUT and VSYNC\* OUT pins.

The CR0\_5 command bit specifies whether a 0 or 7.5 IRE blanking pedestal is to be used. Command bits CR0\_4, CR0\_3, and CR0\_2 specify whether the RGB outputs contain sync information.

### ***SENSE\* Output***

**SENSE\*** is a logical zero if one or more of the IOR, IOG, or IOB outputs have exceeded the internal voltage reference level of the **SENSE\*** comparator circuit. This output determines the presence of a CRT monitor and, with diagnostic code, the difference between a loaded or an unloaded RGB line can be discerned. The reference is generated by a voltage divider from the external 1.235 V voltage reference on the VREF pin. For the proper operation of the **SENSE** circuit, the following levels should be applied to the comparator with the IOR, IOG, and IOB outputs:

DAC Low Voltage  $\leq 260$  mV (see note below)

DAC High Voltage  $\geq 410$  mV (see note below)

There is an additional  $\pm 10\%$  tolerance on the above levels when the internal voltage reference is used.

C/HSYNC\* should be a logical zero and BLSNK\* should be a logical one for SENSE\* to be stable. The SENSE\* output can drive only one CMOS load.

Note: SENSE values are subject to change upon completion of characterization.

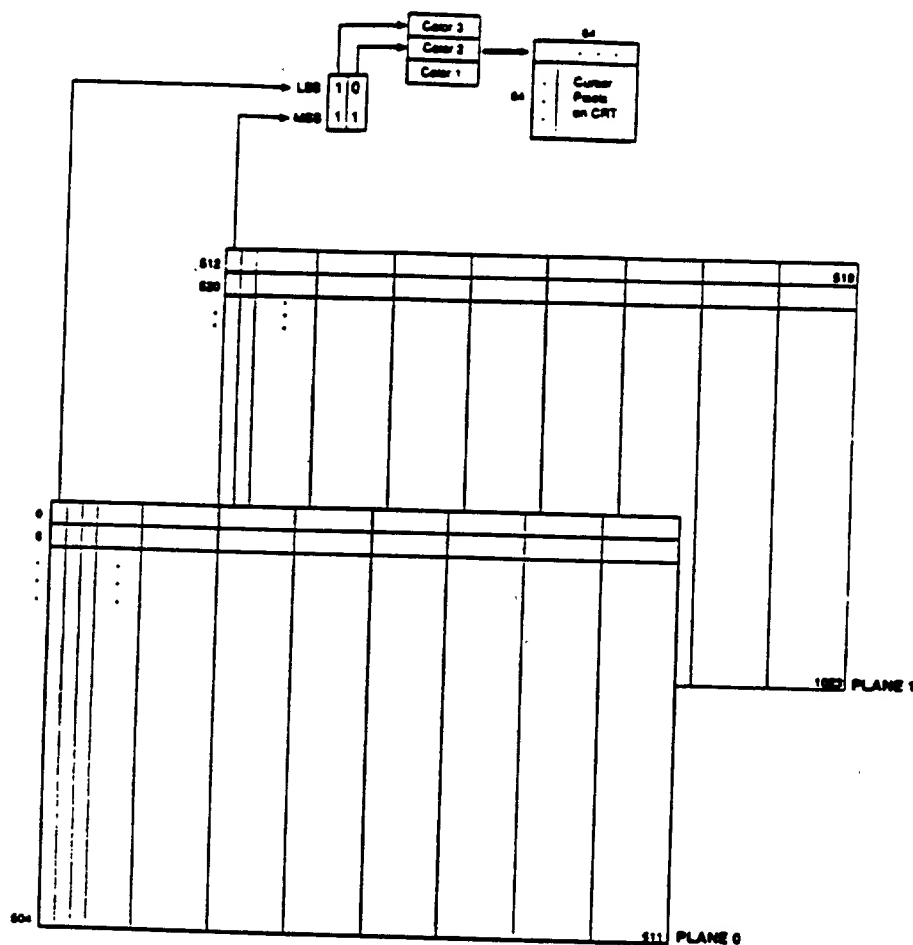
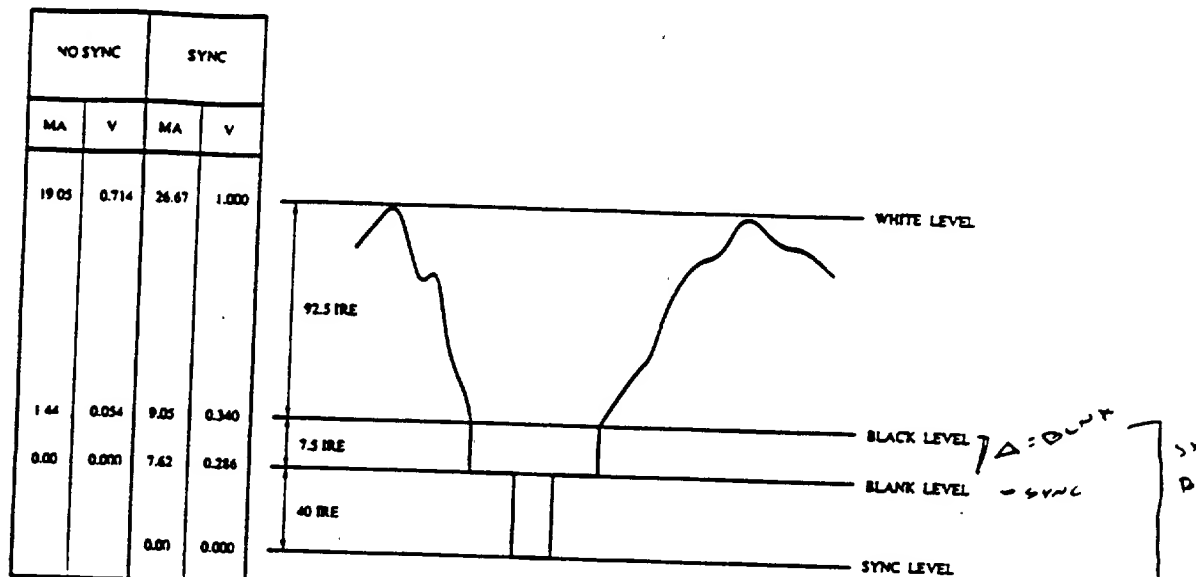


Figure 4. Planar Pixel Format and Cursor RAM Array Pixel Mapping.

Plane 1	Plane 0	MODE 1	MODE 2	MODE 3
0	0	Cursor Not Displayed	Cursor Color 1	Palette Data
0	1	Cursor Color 1	Cursor Color 2	Palette Data
1	0	Cursor Color 2	Palette Data	Cursor Color 1
1	1	Cursor Color 3	Highlight	Cursor Color 2

Table 11. Cursor Color Modes.

Circuit Description (continued)



Note: 75  $\Omega$  doubly terminated load, VREF = 1.235 V, and RSET = 147  $\Omega$ . RS-343A levels and tolerances assumed on all levels.

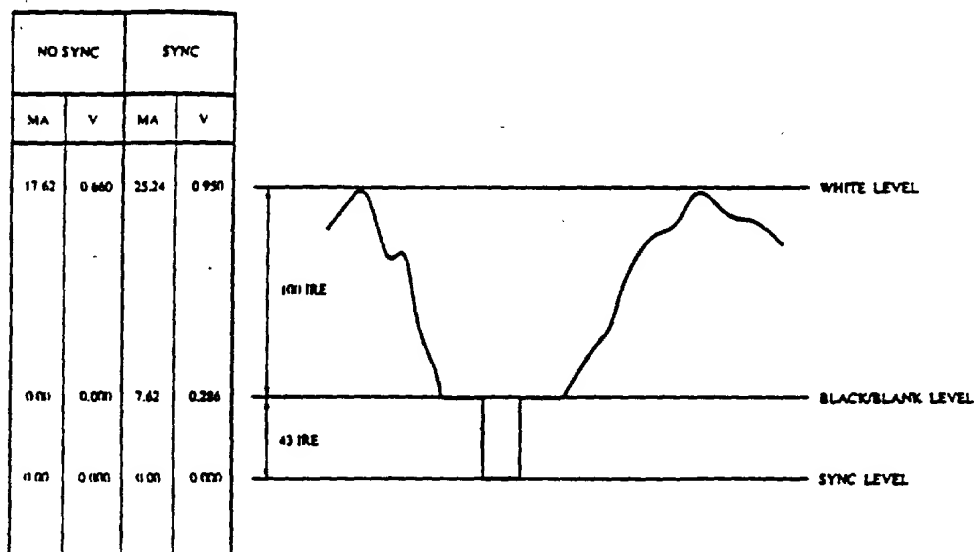
Figure 5. Composite Video Output Waveforms (SETUP = 7.5 IRE).

Description	Sync Disabled	Sync Enabled	C/HSYNC <sup>*</sup>	BLANK <sup>*</sup>	DAC Input Data
	Iout (mA)	Iout (mA)			
WHITE	19.05	26.67	1	1	0xFF
DATA	Data + 1.44	Data + 9.05	1	1	Data
DATA-SYNC	Data + 1.44	Data + 1.44	0	1	Data
BLACK	1.44	9.05	1	1	0x00
BLACK-SYNC	1.44	1.44	0	1	0x00
BLANK	0	7.62	1	0	xx
C/HSYNC <sup>*</sup>	0	0	0	0	xx

Note: 75  $\Omega$  doubly terminated load, VREF = 1.235 V, and RSET = 147  $\Omega$ .

Table 12. Video Output Truth Table (SETUP = 7.5 IRE).

## Circuit Description (continued)



Note: 75  $\Omega$  doubly terminated load, VREF = 1.235 V, and RSET = 147  $\Omega$ . RS-343A levels and tolerances assumed on all levels.

Figure 6. Composite Video Output Waveforms (SETUP = 0 IRE).

Description	Sync Disabled	Sync Enabled	C/HSYNC*	BLANK*	DAC Input Data
	Iout (mA)	Iout (mA)			
WHITE	17.62	25.24	1	1	0xFF
DATA	Data	Data + 7.62	1	1	Data
DATA-SYNC	Data	Data	0	1	Data
BLACK	0	7.62	1	1	0x00
BLACK-SYNC	0	0	0	1	0x00
BLANK	0	7.62	1	0	xx
C/HSYNC	0	0	0	0	xx

Note: 75  $\Omega$  doubly terminated load, VREF = 1.235 V, and RSET = 147  $\Omega$ .

Table 13. Video Output Truth Table (SETUP = 0 IRE).

## Internal Registers

## Command Register 0

This register may be written to or read by the MPU at any time and is not initialized at power-up. CR0\_0 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET<sup>®</sup> pin.

CR0_7	Reserved	This bit must be written with a 0 to ensure proper operation.
CR0_6	Clock Disabled and with CR00 (0) Normal (1) Disable Internal Clocking	When this bit and CR0_0 are a logical one, the internal clock and output clocks are disabled to further conserve power when in power-down mode. The RAM still retains the data, and MPU reads and writes can occur with no loss of data. When this bit is a logical zero, internal clocking is enabled and output clocks will be generated.
CR0_5	Pedestal IRE (0) Disable (1) Enable 7.5 IRE	This bit determines the video blanking pedestal. A logical zero always sets a 0 IRE blanking pedestal and a logical one sets 7.5 IRE.
CR0_4	Blue Sync Enable	These bits specify whether the respective IOB, IOG, or IOR outputs are to contain sync information.
CR0_3	Green Sync Enable	
CR0_2	Red Sync Enable (0) Disable (1) Enable Sync	
CR0_1	DAC 6/8-Bit Resolution (0) 6-Bit Operation (1) 8-Bit Operation	This bit specifies whether the MPU is reading and writing 8 bits (logical one) or 6 bits (logical zero) of color information each cycle.
CR0_0	Power-Down Enable (0) Normal Operation (1) Power-Down Operation	While this bit is a logical zero, the device operates normally. If this bit is a logical one, the DACs and power to the RAM and VideoCache™ FIFO are turned off. The RAM still retains the data, and CPU reads and writes can occur with no loss of data. The DACs will be turned off during sleep mode only if a voltage reference (internal or external) is used.

## Internal Registers (continued)

## Command Register 1

This register may be written to or read by the MPU at any time. CR1\_0 corresponds to data bus bit D0, the least significant data bit (see Table 14). All command register bits are set to logical zero upon asserting a low signal on the RESET\* pin.

CR1_7	CR1_6	CR1_5	CR1_4	Pixel Latching Sequence	Bytes Per VLCLK	Pixels Per VLCLK	Operating Modes
0	0	0	0	N/A	N/A	N/A	All Video Modes Disabled
0	0	0	1	P7:0(H)	1	2	CCIR601 YCrCb 1:0.5:0.5
0	0	1	0	P7:0(H) P7:0(G)	2	4	CCIR601 YCrCb 1:0.5:0.5
0	0	1	1	P7:0(H) P7:0(G) P7:0(F) P7:0(E)	4	8	CCIR601 YCrCb 1:0.5:0.5
0	1	0	0	P7:0(H)	1	1	CCIR601 YCrCb 2:1:1
0	1	0	1	P7:0(H) P7:0(G)	2	2	CCIR601 YCrCb 2:1:1
0	1	1	0	P7:0(H) P7:0(G) P7:0(F) P7:0(E)	4	4	CCIR601 YCrCb 2:1:1
0	1	1	1	P7:0(H-G)	2	1	CCIR601 YCrCb 4:2:2
1	0	0	0	P7:0(H-G) P7:0(F-E)	4	2	CCIR601 YCrCb 4:2:2
1	0	0	1	P7:0(H-G)	2	1	15 Bits Per Pixel, 5:5:5
1	0	1	0	P7:0(H-G) P7:0(F-E)	4	2	15 Bits Per Pixel, 5:5:5
1	0	1	1	P7:0(H-G)	2	1	16 Bits Per Pixel, 5:6:5
1	1	0	0	P7:0(H-G) P7:0(F-E)	4	2	16 Bits Per Pixel, 5:6:5
1	1	0	1	P7:0(H-F)	3	1	24 Bits Per Pixel
1110-1111							Reserved

Table 14. Modes of Operation (Video Pixel Port Configuration).



## Internal Registers (continued)

A. Command register bits are set to logical zero upon asserting a low signal on the RESET\* pin (see Table 15)

CR1_3	CR1_2	CR1_1	CR1_0	Pixel Latching Sequence	MUX Rate	Operating Modes
0	0	0	0	P7:0(A)	1:1	VGA 8 Bits per pixel
0	0		1	P7:0(A) P7:0(B)	2:1	8 Bits per pixel
0	0	1	0	P7:0(A) P7:0(B) P7:0(C) P7:0(D)	4:1	8 Bits per pixel
0	0	1	1	P7:4(A) P3:0(A) P7:4(B) P3:0(B) P7:4(C) P3:0(C) P7:4(D) P3:0(D)	8:1	4 Bits per pixel
0	1	0	0	P7:0(B-A)	1:1	15 Bits Per Pixel, 5:5:5
0	1	0	1	P7:0(B-A) P7:0(D-C)	2:1	15 Bits Per Pixel, 5:5:5
0	1	1	0	P7:0(B-A)	1:1	16 Bits Per Pixel, 5:6:5
0	1	1	1	P7:0(B-A) P7:0(D-C)	2:1	16 Bits Per Pixel, 5:6:5
1	0	0	0	P7:0(C-A)	1:1	24 Bits Per Pixel
1001-1111						Reserved

Table 15. Modes of Operation (Graphic Pixel Port Configuration).

## Internal Registers (continued)

## Command Register 2

This register may be written to or read by the MPU at any time. CR2\_0 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET\* pin.

CR0_7	Reserved Logical 0	This bit must be written with a 0 to ensure correct operation.
CR0_6	Reserved Logical 0	This bit must be written with a 0 to ensure correct operation.
CR0_5	True-Color Bypass Enable (0) Pixel Addresses Palette (1) Pixel Bypasses Palette	When this bit is a logical zero, the pixel palette is addressed by the pixel data. When this bit is a logical one, the RGB pixel data bypasses the color palette and drives the DACs directly. True-color bypassing is only available for pixel sizes of 16 and 24 bits.
CR0_4	Oscillator Select (0) OSC Selected (1) OSC* Selected	When this bit is a logical zero, OSC is selected as the TTL pixel clock input. When this bit is a logical one, OSC* is selected as the TTL pixel clock input.
CR0_3	Display Mode Select (0) Noninterlaced (1) Interlaced	When this bit is a logical zero, the display format is noninterlaced. When the bit is a logical one, the display format is interlaced. The mode must be set properly to ensure proper operation of the internal cursor.
CR0_2	16-Bit/Pixel Palette Index Select (0) Sparse Indexing (1) Contiguous Indexing	When this bit is a logical zero, palette addressing is sparse. The RGB color component pixel data is mapped to the most significant bits of the RGB palette address. The LSBs of the palette address bits are set to (0). When this bit is a logical one, palette addressing is contiguous. The RGB color component pixel data is mapped to the LSBs of the palette address. The MSBs of the address are set to (0).
CR0_1 CR0_0	Cursor Mode Select (00) Cursor Disabled (01) Three-Color Cursor (10) Two-Color/Microsoft Windows™ Cursor (11) Two-Color/X-Windows Cursor	These bits determine the functionality of the onboard 64 x 64 x 2 hardware cursor.

## Internal Registers (continued)

**Accessing the Extended Registers**

An extended register set is used to accommodate all features of the Bt885. Since there are only four register select lines (and all 16 combinations have already been used), the extended registers must be accessed indirectly.

For example, Command Register 3 is accessed with the following sequence of operations:

1. Set RS3–RS0 = 0000, Address Register.
2. Write Address Register to 0x02.
3. Set RS3–RS0 = 1010 (Extended Address Register).
4. Read or Write Command Register 3.

Table 16 shows the indirect addressing mapping for each extended register.

Address Register Value	Extended Register Name
0x00	Status Register 1 (read only)
0x01	Status Register 2 (read/write)
0x02	Command Register 3
0x03	Command Register 4
0x04 – 0x05	Video Window XSTART—Low and High
0x06 – 0x07	Video Window YSTART—Low and High
0x08 – 0x09	Video Window XWIDTH—Low and High
0x0A – 0x0B	Video Window YHEIGHT—Low and High
0x0C – 0x0D	Reserved
0x0E – 0x0F	Reserved
0x10 – 0x11	ESCALEINT—Low and High
0x12 – 0x13	XSCALEINC—Low and High
0x14 – 0x15	Reserved
0x16 – 0x17	Reserved
0x18 – 0x19	Serial Clock Enable Start (Horizontal)—Low and High
0x1A – 0x1B	Serial Clock Enable Duration (Horizontal)—Low and High
0x1C – 0x1D	Reserved
0x1E – 0x1F	Reserved
0x20	DIVCLK1 Rate
0x21	DIVCLK2 Rate
0x22	Reserved
0x23 – 0x25	Color Mask (Ordering = RGB)
0x26	Reserved
0x27 – 0x29	Color Key (Ordering = RGB)
0x2A – 0x2D	Reserved
0x2E	VideoCache™ FIFO Size
0x2F – 0xFF	Reserved

Table 16. Extended Registers Address Map (RS3–RS0 = 1010).

## Internal Registers (continued)

## Command Register 3

This register may be written to or read by the MPU at any time. CR3\_0 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET\* pin.

CR3_7	MODE0 Input/Output Select (0) MODE0 Input (1) MODE0 Output	This bit determines if the MODE0 pin is configured as an input or an output.
CR3_6	Enable Internal Load Clock (0) Use GLCLK (1) Use Internal DIVCLK2	In applications where an external load clock is not provided, setting CR3_6 = 1 allows the internal DIVCLK2, determined by the DIVCLK2 Register values, to internally sample the graphics input pixels, blanking, horizontal, and vertical sync inputs. Setting CR3_6 = 0 causes Bt885 to sample these inputs on the basis of GLCLK pin.
CR3_5	DIVCLK2 Select (0) DIVCLK2 Enabled (1) DIVCLK2 Disabled	A logical zero must be written to this bit to enable the graphics divide-down clock, DIVCLK2, to be output. A logical one written to this bit three-states the DIVCLK2 output.
CR3_4	ECL Clock Select (0) TTL Level Clock Selected (1) Differential ECL Level Clock Selected	A logical one written to this bit enables the differential ECL clock input buffer using OSC and OSC* as inputs. A logical zero written to this bit disables the ECL clock buffer and allows OSC, GLCLK, or the 2x clock multiplier to directly drive the logic. If a logical one is written to this bit, then the clock multiplier and TTL clock selections are overridden. If CR3_4 = 1, then bit CR3_3 must be set to zero.
CR3_3	2x Clock Multiplier Select (0) 2x Clock Multiplier Disabled (1) 2x Clock Multiplier Enabled	This bit enables or disables the 2x clock multiplier. A logical one written to this bit enables the onboard 2x TTL clock multiplier for high-speed operations. A logical zero written to this bit will disable the clock multiplier and will allow the external clock source to directly drive the logic. If CR3_4 = 1, then this bit must be set to zero.
CR3_2	DIVCLK1 Select (0) DIVCLK1 Enabled (1) DIVCLK1 Disabled	A logical zero must be written to this bit to enable the video divide-down clock, DIVCLK1, to be output. A logical one written to this bit three-states the DIVCLK1 output. If DIVCLK1 Select is set to one, then the SEN output pin is three-stated as well.
CR3_1, CR3_0	MSBs for 10-Bit Address Counter CR3_1 = A9 CR3_0 = A8	CR3_1 and CR3_0 are 2 MSBs of the 10-bit cursor address counter. To set this counter to access a particular location in the 64 x 64 x 2 cursor RAM array, these 2 bits must be written to Command Register 3 before the lower 8 bits are written to the address counter through the MPU port. As the 10-bit address counter autoincrements, the new values of this register can be read back through CR3_1 and CR3_0.

## Internal Registers (continued)

## Command Register 4

This register may be written to or read by the MPU at any time and is not initialized at power-up. CR4\_0 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET\* pin.

CR4_7	Video Cache™ FIFO Reset (0) Normal Operation (1) reset VideoCache™ FIFO	A logical zero written to this bit, enables normal VideoCache™ FIFO operation. A logical one written to this bit resets the VideoCache™ FIFO after four video load clocks.
CR4_6	Color Key Override (0) Normal Color Key Operation (1) Video Window Override	A logical zero written to this bit, enables standard color key operation. A logical one written to this bit enables video based only on the video window.
CR4_5	Set MODE0 State (CR3_7 = 0) (0) MODE0 Pin Low (1) MODE0 Pin High	When CR3_7 = 1, this bit controls the state of the MODE0 output. A logical one written to this bit sets the MODE0 pin to high. A logical zero written to this bit sets the MODE0 pin low.
	Get MODE0 State (CR3_7 = 0) (0) MODE0 Pin Externally Driven Low (1) MODE0 Pin Externally Driven High	When CR3_7 = 0, this bit indicates the state of the MODE0 input. A logical one read from this bit indicates that the MODE0 pin is driven high. A logical one read from this bit indicates that the MODE0 pin is driven high. A logical zero read from this bit indicates that the MODE0 pin is driven low.
CR4_4	24-Bit Video Component Order (0) Before Palette (1) After Palette	A logical one written to this bit enables the differential ECL clock input buffer using OSC and OSC* as inputs. A logical zero written to this bit disables the ECL clock buffer and allows OSC, GLCLK, or the 2x clock multiplier to directly drive the logic. If a logical one is written to this bit, then the clock multiplier and TTL clock selections are overridden. If CR3_4 = 1, then bit CR3_3 must be set to zero.
CR4_3	Color Key Mode Select (0) Before Palette (1) After Palette	This bit controls whether color key matching occurs on the pixel value before or after the palette. A logical zero written to this bit selects color key matching on the pixel value before the palette. A logical one written to this bit selects color key matching on the 24-bit RGB value after the palette.
CR4_2	Video Cache™ Unload Select (0) Unload Within Video Window (1) Unload From Start of Active Graphic Enable	This bit controls whether VideoCache™ FIFO data is unloaded only within the video window or at all times during active graphics enable.
CR4_1, CR4_0	24-Bit Graphics Component Order (00) RGB (01) BRG (10) BGR (11) Reserved	This bit controls the component latching order in 24-bit-per-pixel graphic modes. If any other graphics mode is selected, these bits must be set to logical zero.

## Internal Registers (continued)

**Pixel Read Mask Register**

The 8-bit pixel read mask register may be written to or read by the MPU at any time, and is initialized to 0xFF at power-up. D0 is the least significant bit. The contents of this register are bit-wise ANDed with the pixel data prior to addressing the color palette RAM.

**Status Registers 1 and 2**

These two 8-bit status registers are provided for device identification and to monitor certain device states. They may be read by the MPU at any time. MPU write cycles to status register 1 are ignored. D0 is the least significant bit corresponding to SR1\_0 or SR2\_0. These registers are not reset during power-up/reset.

SR1_7, SR1_6	Chip Identification	These bits are identification values; SR1_7 = 1 and SR1_6 = 0.
SR1_5, SR1_4	Chip Revision	These bits are revision values; SR1_5 = 1 and SR1_4 = 0.
SR1_3	Monitor Sense	This is the SENSE* bit. If it is a logical zero, one or more of the IOR, IOG, and IOB outputs have exceeded the internal voltage reference level (335 mV). This bit is used to determine the presence of a CRT monitor and, with diagnostic code, the difference between a loaded or unloaded RGB line can be discerned. The 360 mV reference has a $\pm 100$ mV tolerance when an external voltage reference equal to 1.235 V is used. A greater tolerance is expected when an internal reference equal to 1.2 V is used.
SR1_2	Read/Write Access Status (0) Write Cycle (1) Read Cycle	This bit provides RD/WR status when Address Register 0x00, 0x03, 0x04, or 0x07 has been written. When Address Register 0x00 or 0x04 has been written, the device is in the write mode and this bit is a logical zero. When address register 0x03 or 0x07 has been written, the device is in the read mode and this bit is a logical one.
SR1_1, SR1_0	RGB Component Counter (00) Red Color Component (01) Green Color Component (10) Blue Color Component	When read, these bits reflect the color component address for the next RD/WR cycle when accessing the palette, cursor color registers, or overscan register.
SR2_7	Video Cache™ FIFO Underflow	Reading this bit as a one indicates that VideoCache™ FIFO underflow occurred. Reset by writing any value to Status Register 2.
SR2_6-0	Reserved	These bits will always be read as zero.

## Internal Registers (continued)

### Video Window XSTART

Video Window XSTART is a 12-bit register that stores the starting X position on the screen for a video window. A value of zero indicates that the video window begins in the first (leftmost) pixel of each horizontal scan line.

### Video Window YSTART

Video Window YSTART is a 12-bit register that stores the starting Y position on the screen for a video window. A value of zero indicates that the video window begins on the first active graphics scan line.

### Video Window XWIDTH

Video Window XWIDTH is a 12-bit register that stores the number of pixels per scan line within the video window. A value of zero indicates that no pixels are in the video window.

### Video Window YHEIGHT

Video Window YHEIGHT is a 12-bit register that stores the number of scan lines within the video window. A value of zero indicates that no scan lines are within the video window.

### XSCALEINIT (Low and High)

XSCALEINIT are 12-bit registers that store the initial term for the horizontal scaler.

### XSCALEINC (Low and High)

XSCALEINC are 12-bit registers that store the increment term for the horizontal scaler.

### Serial Clock Enable Start (Horizontal)

Serial clock enable start (horizontal and vertical) are 12-bit registers that store the number of scan lines and DIVCLK1 cycles before enabling the external clock gate, starting at the leading edge of HSYNC\* for the horizontal direction and the leading edge of the internally generated VSYNC\* for the vertical direction.

### Serial Clock Enable Duration (Horizontal)

Serial clock enable duration (horizontal and vertical) are 12-bit registers that store the number of serial shift clock cycles to be generated per scan line in units of DIVCLK1 cycles for the horizontal direction, and in units of scan lines for the vertical direction.

### DIVCLK1 and DIVCLK2 Rate

DIVCLK1 and DIVCLK2 rate are two 3-bit registers that control the divide rate of the free-running DIVCLK1 and DIVCLK2 divide-down clocks, respectively. The divide-down ratios need not be the same as the input mux rate:

- (000) - 1:1
- (001) - 2:1
- (010) - 4:1
- (011) - 8:1
- (100-111) - Reserved

### VideoCache™ FIFO Size

This register indicates the length of the VideoCache™ FIFO buffer in 16-byte units. This is a read-only register.

## Internal Registers (continued)

**Cursor (x,y) Registers**

These registers are used to specify the (x,y) coordinate of the 64 x 64 x 2 hardware cursor. The cursor (x) register is made up of the cursor (x) low register (CXLR) and the cursor (x) high register (CXHR); the cursor (y) register is made up of the cursor (y) low register (CYLR) and the cursor (y) high register (CYHR). The last value written by the MPU to these registers is the value returned on a read. These registers may be written to or read by the MPU at any time.

CXLR and CXHR are cascaded to form a 12-bit cursor (x) register. Similarly, CYLR and CYHR are cascaded to form a 12-bit cursor (y) register. Bits D4-D7 of CXHR and CYHR are ignored and should be written as zeros.

The cursor (x) value to be written is calculated as follows:

$$Xp = \text{desired display screen (x) position} + 64$$

where the (x) reference point for the display screen,  $x = 0$ , is the upper left corner of the screen. The  $Xp$  position

equation places the upper lefthand corner of the cursor RAM array to the desired screen location. This allows the cursor position to be defined in the same coordinate space as the screen.

Values from 0 to 4095 may be written into the cursor (x) register. If  $Xp$  is equal to zero, the cursor will be entirely offscreen.

The cursor (y) value to be written is calculated as follows:

$$Yp = \text{desired display screen (y) position} + 64$$

where the (y) reference point for the display screen,  $y = 0$ , is the upper left corner of the screen. The  $Yp$  position equation places the upper left corner of the cursor RAM array to the desired screen location. This allows the cursor position to be defined in the same coordinate space as the screen.

Values from 0 to 4095 may be written into the cursor (y) register. If  $Yp$  is equal to zero, the cursor will be entirely offscreen (see Cursor Operation in the Circuit Description section).

Register Values on reset are given in Table 17.

	Cursor (x) High (CXHR)				Cursor (x) Low (CXLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Address	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

	Cursor (y) High (CYHR)				Cursor (y) Low (CYLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Address	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0



## Internal Registers (continued)

Register Name	Reset Value
Command Register 0	0
Command Register 1	0
Command Register 2	0
Command Register 3	0
Command Register 4	0
Video Window XSTART—Low and High	Not Initialized
Video Window YSTART—Low and High	Not Initialized
Video Window XWIDTH—Low and High	Not Initialized
Video Window YHEIGHT—Low and High	Not Initialized
XSCALEINT—Low and High	Not Initialized
XSCALEINC—Low and High	Not Initialized
Serial Clock Enable Start (Horizontal)—Low and High	Not Initialized
Serial Clock Enable Duration (Horizontal)—Low and High	Not Initialized
Serial Clock Enable Start (Vertical)—Low and High	Not Initialized
Serial Clock Enable Duration (Vertical)—Low and High	Not Initialized
DIVCLK1 Rate	0
DIVCLK2 Rate	0
Color Mask	0
Color Key	0
FIFO Size	0x32
Color Palette RAM	Not Initialized
Pixel Read Mask	0xFF
Cursor Colors	Not Initialized
Overscan Color	Not Initialized
Cursor X,Y	Not Initialized
Cursor RAM Array	Not Initialized

Table 17. Register Values on Reset.

## Pin Description

Pin Name	I/O	Pin #	Description												
RESET*	I	72	Reset input (TTL compatible). When this signal is low, all the command register bits are initialized to zero and the device is in VGA mode.												
BLANK*	I	98	Composite blank control input (TTL compatible). A logical zero drives the analog outputs to the blanking level, as specified in Figure 5 and Figure 6. It is latched on the rising edge of GLCLK. When BLANK* is a logical zero, the pixel inputs are ignored.												
ENABLE (Composite Display Enable)	I	96	Composite display enable control input (TTL compatible). The state of this signal and BLANK* determines whether the analog outputs are blanked or contain cursor color, pixel, or overscan data. This signal is latched on the rising edge of GLCLK. If overscanning is not used, this pin should be tied to BLANK*. The following table lists the combinations of ENABLE and BLANK*:												
<table><tr><th>ENABLE</th><th>BLANK*</th><th>Operation</th></tr><tr><td>x</td><td>0</td><td>Video Blanking</td></tr><tr><td>0</td><td>1</td><td>Overscan Data</td></tr><tr><td>1</td><td>1</td><td>Cursor Color or Pixel Data</td></tr></table>				ENABLE	BLANK*	Operation	x	0	Video Blanking	0	1	Overscan Data	1	1	Cursor Color or Pixel Data
ENABLE	BLANK*	Operation													
x	0	Video Blanking													
0	1	Overscan Data													
1	1	Cursor Color or Pixel Data													
ODD/EVEN*	I	95	Odd/even field input (TTL compatible). This signal should be changed only during vertical blank. This input is used to ensure proper operation of the onboard cursor when interlaced operation (command bit CR2_3 = 1) is selected. When this signal is a logical zero, an even field is specified. When this signal is a logical one, an odd field is specified. This input is ignored if noninterlaced operation (command bit CR2_3 = 0) is selected.												
OSC, OSC*	I	131, 132	Pixel clock input (ECL/TTL compatible). This input is an ECL-compatible input, but a TTL clock may be used on either OSC or OSC* if selected by CR2_4 in Command Register 1 (CR3_4 = 0). It is recommended that all clock inputs be driven by a dedicated buffer to avoid reflection-induced jitter. In 1:1 mode DIVCLK = OSC or DIVCLK = OSC*.												
DIVCLK1	O	127	Frame buffer shift clock output (TTL compatible). The signal on this pin is equal to the selected pixel clock divided by 8, 4, 2, or 1, depending on the selection in the DIVCLK1 rate register. This output has low drive capability. DIVCLK1 and DIVCLK2 are opposite phases.												
DIVCLK2	O	128	Frame buffer shift clock output (TTL compatible). The signal on this pin is equal to the selected pixel clock divided by 8, 4, 2, or 1, depending on the selection in the DIVCLK2 rate register. This output has low drive capability. DIVCLK1 and DIVCLK2 are opposite phases.												
FIFO RESET*	I	14	A high value applied to this pin enables normal VideoCache™ FIFO operation. Although FIFO RESET* is level sensitive, a transition from high to low on this pin and should be kept low for at least two VLCLKs in order for the VideoCache™ FIFO to be properly reset.												

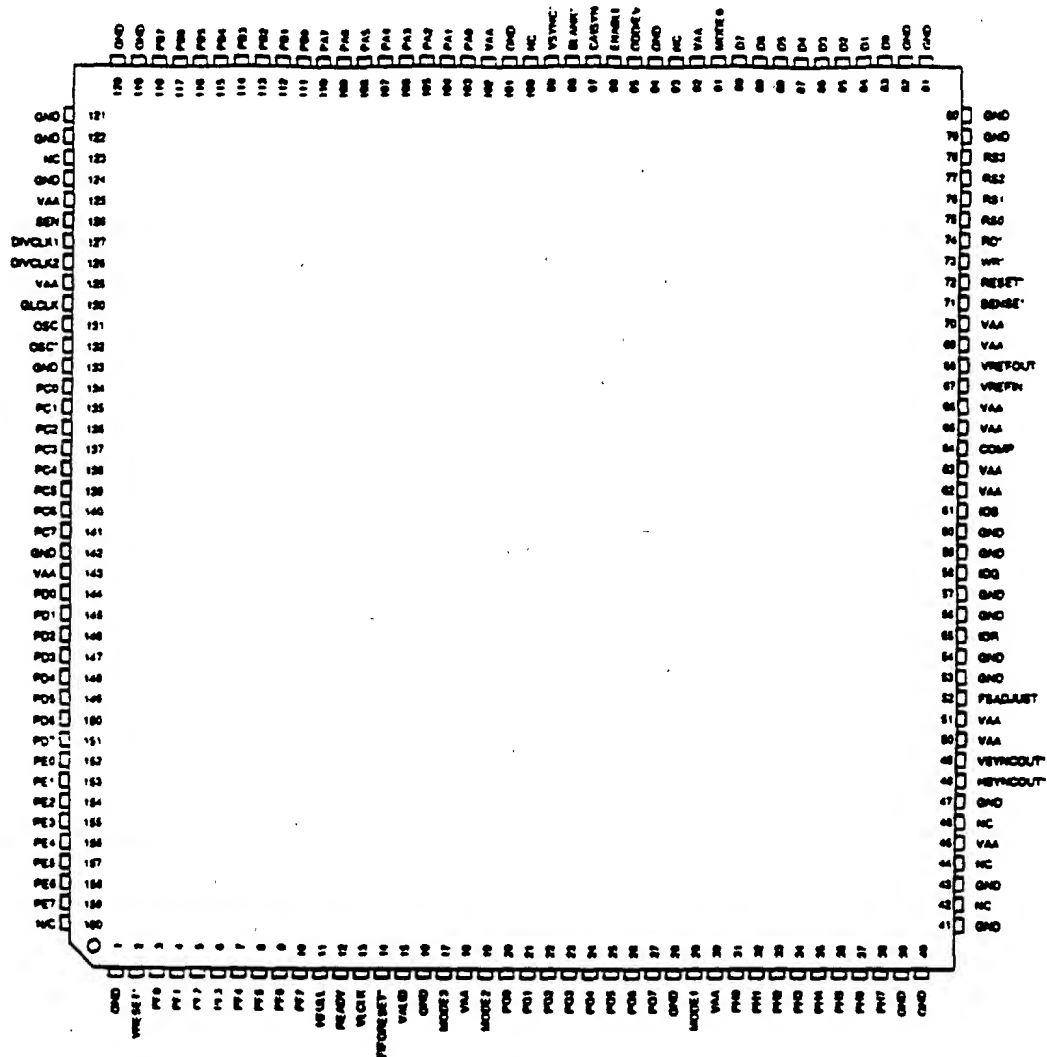
## Pin Description (continued)

Pin Name	I/O	Pin #	Description
GLCLK	I	13	Graphics port input load clock (TTL compatible with hysteresis). The rising edge of this signal latches P7:0 (A-D), BLANK*, ENABLE, HSYNC*, and VSYNC*.
VLCLK	I	13	Video port input load clock (TTL compatible with hysteresis). The rising edge of this signal latches P7:0 (E-H).
P7:0 (A-H)	I	See Pin Diagram	Pixel port inputs (TTL compatible). This port can be used in various modes, as shown in Tables 10 and 11, for video and or graphics input.
VALID	I	15	Video port input pixel data valid signal (TTL compatible).
READY	O	12	Video port input pixel data ready signal (TTL compatible, low drive). This signal can be synchronously sampled using the rising edge of VLCLK. This signal changes only following a rising edge of VLCLK.
HFULL	O	11	VideoCache™ FIFO half-full or greater signal. (TTL compatible, low drive).
SEN	O	126	DIVCLK1 gating control signal (TTL compatible, low drive). It may be used to externally gate the DIVCLK1 output to generate a gated version of DIVCLK1. This signal changes only during DIVCLK1 low duration. The start time and duration of the pulse train may be programmed relative to the leading edge of C/HSYNC* and internally generated VSYNC.
VRESET*	O	2	Vertical reset signal (TTL compatible, low drive). This signal is generated to allow the asynchronous video data to know the start of each frame. This signal is synchronous to VLCLK.
MODE0	I/O	91	General purpose registered input/output (TTL compatible) set or read using CR4_5. Selection of input or output is made using CR3_7. Must be tied high with a 10 K pullup resistor.
MODE1-MODE3	I/O	29, 17, 19	Reserved for future expansion. Must be tied high with a 10 K pullup resistor.
WR*	I	73	Write control input (TTL compatible with hysteresis). D0-D7 data is latched on the rising edge of WR*, and RS0-RS3 are latched on the falling edge of WR* during MPU write operations. RD* and WR* should not be asserted simultaneously.
RD*	I	74	Read control input (TTL compatible with hysteresis). To read data from the device, RD* must be a logical zero. RS0-RS3 are latched on the falling edge of RD* during MPU read operations. RD* and WR* should not be asserted simultaneously.
RS0-RS3	I	75-78	Register select inputs (TTL compatible). RS0-RS3 specify the type of read or write operation being performed, as specified in Table 1 and Table 2.
D0-D7	I/O	83-90	Data bus (TTL compatible). data is transferred into and out of the device over this 8-Bit bidirectional data bus. D0 is the least significant bit.
SENSE*	O	71	Comparator sense output (CMOS compatible). This pin will be low if one or more of the IOR, IOG, and IOB analog output levels is above the internal comparator reference of 350 mV $\pm$ 50 mV.
IOR, IOG, IOB	A, O	55, 58, 61	Red, green, and blue current outputs. These high-impedance current sources can directly drive a doubly terminated 75 $\Omega$ coaxial cable (see the PC Board Layout Considerations section for further information).

## Pin Description (continued)

Pin Name	I/O	Pin #	Description																				
C/HSYNC*	I	97	Horizontal or composite sync control input (TTL compatible).																				
VSYNC*	I	99	Vertical sync control input (TTL compatible). This signal is pipelined to VSYNCOUT*.																				
HSYNCOUT*, VSYNCOUT*	O	48, 49	Pipeline delayed horizontal and vertical sync control signals.																				
FSADJUST	A, I	52	Full-scale adjust control. The IRE relationships in Figure 5 and Figure 6 are maintained, regardless of the full-scale output current.  When an external or the internal voltage reference is used (see Figure 7 and Figure 8 in the PC Board Layout Considerations section), a resistor (RSET) connected between this pin and GND controls the magnitude of the full-scale video signal. The relationship between RSET and the full-scale output current on each output is:  $RSET (\Omega) = K \cdot 1,000 \cdot VREF (V) / I_{out} (mA)$  K is defined in the table below. It is recommended that a 147 $\Omega$ RSET resistor be used for doubly terminated 75 $\Omega$ loads (i.e., RS-343A applications).																				
			<table><tr><td></td><td colspan="2">Sync Enabled</td><td colspan="2">Sync Disabled</td></tr><tr><td>Setup</td><td>0 IRE</td><td>7.5 IRE</td><td>0 IRE</td><td>7.5 IRE</td></tr><tr><td>K (8-bit)</td><td>2.888</td><td>3.055</td><td>2.045</td><td>2.207</td></tr><tr><td>K (6-bit)</td><td>3.000</td><td>3.170</td><td>2.100</td><td>2.260</td></tr></table> K values are subject to change upon completion of characterize		Sync Enabled		Sync Disabled		Setup	0 IRE	7.5 IRE	0 IRE	7.5 IRE	K (8-bit)	2.888	3.055	2.045	2.207	K (6-bit)	3.000	3.170	2.100	2.260
	Sync Enabled		Sync Disabled																				
Setup	0 IRE	7.5 IRE	0 IRE	7.5 IRE																			
K (8-bit)	2.888	3.055	2.045	2.207																			
K (6-bit)	3.000	3.170	2.100	2.260																			
VREFOUT	A, O	68	Voltage reference output. This output provides a 1.2 V (typical) reference and may be connected directly to the VREFIN pin. If the on-chip reference is not used, this pin may be left floating. See Figure 7 and Figure 8.																				
VREFIN	A, I	67	Voltage reference input. If an external voltage reference is used (Figure 8), it must supply this input with a 1.2 V (typical) reference. A 0.1 $\mu$ F ceramic capacitor must be used to decouple this input to GND, as shown in Figure 7 and Figure 8. The decoupling capacitor must be as close to the device as possible to keep lead lengths to an absolute minimum. When the internal reference is used, this pin should not drive any external circuitry, other than the decoupling capacitor (Figure 7).																				
COMP	A, O	64	Compensation pin. A 0.1 $\mu$ F ceramic capacitor must be used to bypass this pin to the nearest VAA pin. The COMP capacitor must be as close as possible to the device to keep lead lengths to an absolute minimum. The compensation capacitor must be connected with short wide traces.																				
VAA	A, P	See Pin Diagram	Analog power. All VAA pins must be connected to the same analog power plane.																				
GND	G	See Pin Diagram	Analog ground. All GND pins must be connected to the same common ground plane.																				

Pin Description (continued)



Note: All pins marked NC are reserved for future expansion and *MUST* be left floating.

## PC Board Layout Considerations

### PC Board Considerations

The layout should be optimized for lowest noise on the Bt885 power and ground planes by providing good decoupling. The trace length between groups of VAA and GND pins should be as short as possible to minimize inductive ringing.

A well-designed power distribution network is critical to eliminating digital switching noise. The ground plane must provide a low-impedance return path for the digital circuits. A PC board with a minimum of four layers is recommended, with layers 1 (top) and 4 (bottom) for signals and layers 2 and 3 for ground and power.

### Component Placement

Components should be placed as close as possible to the associated CacheDAC™ pin. Whenever possible, components should be placed so traces can be connected point to point.

The optimum layout enables the Bt885 to be located as close as possible to the power supply connector and the video output connector.

### Ground Planes

For optimum performance, a common digital and analog ground plane is highly recommended.

### Power Planes

Separate digital and analog power planes are recommended. The digital power plane should provide power to all digital logic on the PC board, and the analog power plane should provide power to all Bt885 power pins, VREF circuitry, and COMP. There should be at least a 1/8-inch gap between the digital power plane and the analog power plane.

The analog power plane should be connected to the digital power plane (VCC) at a single point through a ferrite bead, as illustrated in Figure 7 and Figure 8. This bead should be located within 3 inches of the Bt885. The bead provides resistance to switching currents, acting as a resistance at high frequencies. A low-resistance bead should be used, such as Ferroxcube 5659065-3B, TDK HF30ACB321611T, or TDK BF45-4001.

### Device Decoupling

For optimum performance, all capacitors should be located as close as possible to the device, and the shortest possible leads (consistent with reliable operation) should be used to reduce the lead inductance. Chip capacitors are recommended for minimum lead inductance. Radial lead ceramic capacitors may be substituted for chip capacitors and are better than axial lead capacitors for self-resonance. Values are chosen to have self-resonance above the pixel clock.

### Power Supply Decoupling

The best power supply decoupling performance is obtained with a 0.1  $\mu\text{F}$  ceramic capacitor in parallel with a 0.01  $\mu\text{F}$  chip capacitor decoupling each group of VAA pins to GND. The capacitors should be placed as close as possible to the device VAA and GND pins and connected with short, wide traces.

The 10  $\mu\text{F}$  capacitor shown in Figure 7 and Figure 8 is for low-frequency power supply ripple; the 0.1  $\mu\text{F}$  capacitors are for high-frequency power supply noise rejection.

### COMP Decoupling

The COMP pin must be decoupled to VAA, typically with a 0.1  $\mu\text{F}$  ceramic capacitor and optional 15  $\Omega$  resistor. Low-frequency supply noise will require a larger value. The COMP capacitor must be as close as possible to the COMP and VAA pins. A surface-mount ceramic chip capacitor is preferred for minimal lead inductance. Lead inductance degrades the noise rejection of the circuit. Short, wide traces will also reduce lead inductance.

If the display has a ghosting problem, additional capacitance in parallel with the COMP capacitor may help.

### VREF Decoupling

A 0.1  $\mu\text{F}$  ceramic capacitor should be used to decouple this input to GND.

## PC Board Layout Considerations (continued)

**Digital Signal Interconnect**

The digital inputs to the Bt885 should be isolated as much as possible from the analog outputs and other analog circuitry. Also, these input signals should not overlay the analog power plane or analog output signals.

Most of the noise on the analog outputs will be caused by excessive edge rates (less than 3 ns), overshoot, undershoot, and ringing on the digital inputs.

The digital edge rates should not be faster than necessary, as feedthrough noise is proportional to the digital edge rates. Lower-speed applications will benefit from using lower-speed logic (3–5 ns edge rates) to reduce data-related noise on the analog outputs.

Transmission lines will mismatch if the lines do not match the source and destination impedance. This will degrade signal fidelity if the line length reflection time is greater than one-fourth the signal edge time (refer to Brooktree Application Notes AN-11 and AN-12). Line termination or line-length reduction is the solution. For example, logic edge rates of 2 ns require line lengths of less than 4 inches without use of termination. Ringing may be reduced by damping the line with a series resistor (30–300  $\Omega$ ). The RS-select inputs and RD\*/WR\* lines must be verified for proper levels with no ringing, undershoot, or overshoot. Ringing on these lines can cause improper operation.

Radiation of digital signals can also be picked up by the analog circuitry. This is prevented by reducing the digital edge rates (rise/fall time), minimizing ringing with damping resistors, and minimizing coupling through PC board capacitance by routing the digital signals at a 90 degree angle to any analog signals.

The clock driver and all other digital devices must be adequately decoupled to prevent noise generated by the digital devices from coupling into the analog circuitry.

**TTL Clock Interfacing**

The Bt885 requires a pixel clock with monotonic clock edges for proper operation. Impedance mismatch on the pixel clock line will induce reflections on the pixel clock, which may cause erratic operation.

The Pixel Clock Pulse Width High Time and Pixel Clock Pulse Width Low Time minimum specifications (see the AC Characteristics section) must not be violated, or erratic operation can occur.

The pixel clock line must be terminated to prevent impedance mismatch. A series termination of 33–68  $\Omega$  placed at the pixel clock driver may be used, or a parallel termination may be used at the pixel clock input to the CacheDACTM. A parallel termination of 220  $\Omega$  to VCC and 330  $\Omega$  to ground will provide a Thevenin equivalent of a 110  $\Omega$  termination, which is normally sufficient to absorb reflections. The series or parallel resistor values should be adjusted to provide the optimum clock signal fidelity.

**Differential Clock Interfacing**

Termination requirements for differential ECL clock sources will vary depending on the particular clock generator used.

**MPU Control Signal Interfacing**

The Bt885 uses the RD\*, WR\*, and RS lines to determine which MPU accesses will take place. Glitches or ringing on any of these lines may cause improper MPU operation. When a VGA controller with edge rate control is used on these lines, a series termination is not necessary. In non-VGA controller application or in applications where the MPU control signals are daisy chained, a series termination, pull-down resistors, or additional capacitance to ground should be used to prevent glitches that could cause improper MPU accesses.

**Analog Signal Interconnect**

The Bt885 should be located as close as possible to the output connectors to minimize noise pickup and reflections caused by impedance mismatch.

The analog outputs are susceptible to crosstalk from digital lines; digital traces must not be routed under or adjacent to the analog output traces.

To maximize the high-frequency power supply rejection, the video output signals should not overlay the analog power plane.

For maximum performance, the analog video output impedance, cable impedance, and load impedance should be the same. The load resistor connection between the video outputs and GND should be as close as possible to the Bt885 to minimize reflections. Unused analog outputs should be connected to GND.

Analog output video edges exceeding the CRT monitor bandwidth can be reflected, producing cable-length dependent ghosts. Simple pulse filters can reduce high-frequency energy, reducing EMI and noise. The filter impedance must match the line impedance.

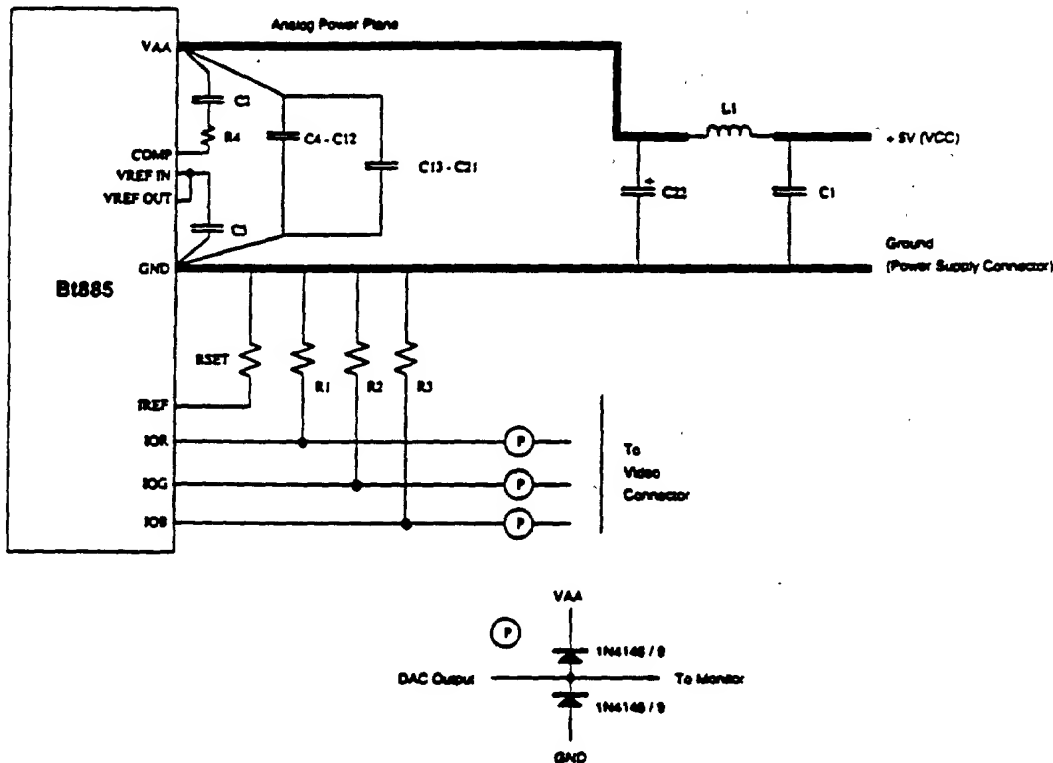
## PC Board Layout Considerations (continued)

## Analog Output Protection

The Bt885 analog output should be protected against high-energy discharges, such as those from monitor arc-over or from hot-switching AC-coupled monitors.

The diode protection circuit shown in Figure 7 and Figure 8 can prevent latchup under severe discharge conditions without adversely degrading analog

transition times. The 1N4148/9 parts are low-capacitance, fast-switching diodes, which are also available in multiple-device packages (FSA250X or FSA270X) or surface-mountable pairs (BAV99 or MMBD7001).

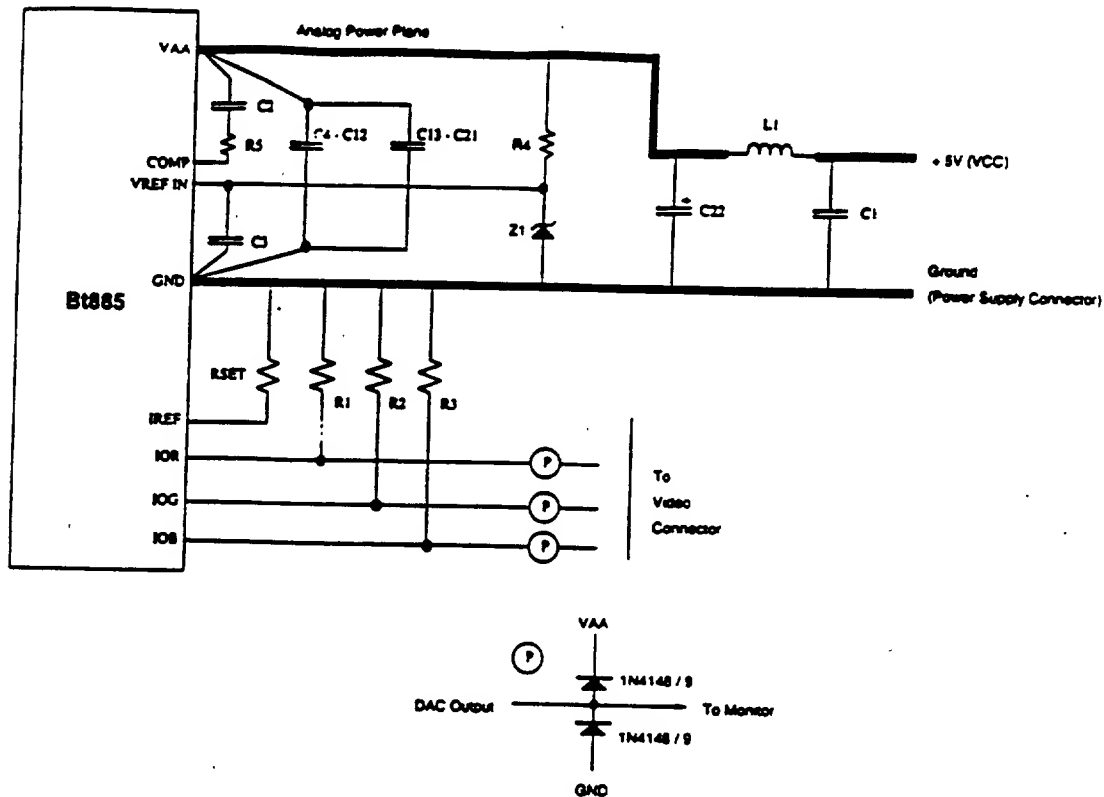


Location	Description	Vendor Part Number
C1-C12	0.1 $\mu$ F Ceramic Capacitor	Erie RPE112Z5U104M50V
C22	10 $\mu$ F Capacitor	Mallory CSR13G106KM
L1	Ferrite Bead	TDK HF30ACB321611T
R1, R2, R3	75 $\Omega$ 1% Metal Film Resistor	Dale CMF-55C
R4	15 $\Omega$ 1% Metal Film Resistor	Dale CMF-55C
RSET	1% Metal Film Resistor	Dale CMF-55C

Figure 7. Typical Connection Diagram and Parts List (Internal Voltage Reference).



PC Board Layout Considerations (continued)



Location	Description	Vendor Part Number
C1-C12	0.1 $\mu$ F Ceramic Capacitor	Erie RPE112Z5U104M50V
C22	10 $\mu$ F Capacitor	Mallory CSR13G106KM
L1	Ferrite Bead	TDK HF30ACB321611T
R1, R2, R3	75 $\Omega$ 1% Metal Film Resistor	Dale CMF-55C
R4	1 K $\Omega$ 5% Metal Film Resistor	Dale CMF-55C
R5	15 $\Omega$ 1% Metal Film Resistor	Dale CMF-55C
RSET	1% Metal Film Resistor	Dale CMF-55C
Z1	1.2 V Voltage Reference	National Semiconductor LM385BZ-1.2

Figure 8. Typical Connection Diagram and Parts List (External Voltage Reference).

**PC Board Layout Considerations (continued)*****Using Multiple Devices***

When multiple Bt885s are used, each Bt885 should have its own power plane and ferrite bead. If the internal reference is used, each Bt885 should use its own internal reference.

Although the multiple Bt885s may be driven by a common external voltage/current reference, higher performance may be obtained if each CacheDAC™ uses its own reference. This will reduce the amount of color channel crosstalk and color palette interaction.

Each Bt885 must still have its own RSET resistor, analog output termination resistors, power supply bypass capacitors, COMP capacitor, and reference capacitors.

***ESD and Latchup Considerations***

Correct ESD-sensitive handling procedures are required to prevent device damage, which can produce symptoms of catastrophic failure or erratic device behavior with leaky inputs.

All logic inputs should be held low until power to the device has settled to the specified tolerance. DAC power decoupling networks with large time constants should be avoided. They could delay VAA power to the device. Ferrite beads must be used only for analog power VAA decoupling. Inductors and regulators cause a time constant delay that induces latchup.

Latchup can be prevented by ensuring that all VAA and GND pins are at the same potential and that the VAA supply voltage is applied before the signal pin voltages. The correct power-up sequence ensures that any signal pin voltage will never exceed the power supply voltage by more than + 0.5 V.

***Sleep Operation***

When the internal or external voltage reference is used, the DACs will be turned off during sleep mode.

When an external voltage reference is used, some internal circuitry will still be powered during the sleep mode. This unnecessary current drain can be disabled by turning off the external voltage reference during power-down mode.

## Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Power Supply	VAA	4.75	5.00	5.25	V
Ambient Operating Temperature	TA	0		-70	°C
Output Load	RL		37.5		$\Omega$
Voltage Reference Configuration Reference Voltage	VREF	1.1112	1.235	1.359	V

## Absolute Maximum Ratings

Parameter	Symbol	Min	Typ	Max	Units
VAA (Measured to GND)				7.0	V
Voltage on Any Signal Pin (Note 1)		GND -0.5		VAA + 0.5	V
Analog Output Short Circuit Duration to Any Power Supply or Common	ISC		Indefinite		
Storage Temperature	TS	-65		+150	°C
Junction Temperature	TJ			+150	°C
Vapor Phase Soldering (1 Minute)	TVSOL			220	°C

Stresses above those listed under "Absolute Maximum Rating," may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Note 1** This device employs high-impedance CMOS devices on all signal pins. It should be handled as an ESD-sensitive device. Voltage on any signal pin that exceeds the power supply voltage by more than + 0.5 V can cause destructive latchup.

## DC Characteristics

Parameter	Symbol	Min	Typ	Max	Units
Resolution (Each DAC) Accuracy (Each DAC) Integral Linearity Error Differential Linearity Error Gray-Scale Error Monotonicity Coding	IL DL	8	8  Guaranteed	8  ±1 ±1 ±5	LSB LSB % Gray Scale Binary
Digital Inputs Input High Voltage Input Low Voltage Input High Current (Vin = 2.4 V) $I_{IH}$ Input Low Current (Vin = 0.4 V) $I_{IL}$ Input Capacitance (V = 1 MHz, Vin = 2.4 V) Hysteresis	V <sub>IH</sub> V <sub>IL</sub> I <sub>IH</sub> I <sub>IL</sub> C <sub>IN</sub>	2.0 GND - 0.5	    0.3	V <sub>AA</sub> + 0.5 0.8 Z <sub>oo</sub> tbd 90 tbd 7	V V μA μA pF V
OSC/OSC* ECL Differential Inputs Input High Voltage Input Low Voltage	Δ Vin V <sub>IH</sub> V <sub>IL</sub>	0.6 V <sub>CC</sub> - 1.1 V <sub>CC</sub> - 2		V <sub>CC</sub> - 0.8 V <sub>CC</sub> - 1.5	V V V
Digital Outputs Output High Voltage (I <sub>OH</sub> = -400 μA) Output Low Voltage (I <sub>OL</sub> = 3.2 mA) Three-State Current Output Capacitance Load Capacitance	V <sub>OH</sub> V <sub>OL</sub>  I <sub>OZ</sub> C <sub>DOUT</sub> CL	2.4		0.4  10 7 10	V V μA pF pF

See test condition and notes on next page.

## DC Characteristics (continued)

Parameter	Symbol	M	Typ	Max	Units
Analog Outputs					
Gray-Scale Current Range				20	mA
Output Current (Standard RS-343A)					
White Level Relative to Black		15.86	17.62	18.5	mA
Black Level Relative to Black					
SETUP = 7.5 IRE		0.95	1.44	1.90	mA
SETUP = 0 IRE		-50	5	50	$\mu$ A
Blank Level		-6.29	7.62	8.96	mA
Sync Level		-50	5	50	$\mu$ A
LSB Size			69.1		$\mu$ A
DAC-to-DAC Matching			2	5	%
Output Compliance	VOC	-0.2		+1.5	V
Output Impedance	RAOUT		10		k $\Omega$
Output Capacitance	CAOUT			30	pF
(f = 1 MHz, IOUT = 0 mA)					
Onboard VREF (Note 1)	VREFOUT	tbd	tbd	tbd	V
Voltage Reference Input Current	I <sub>VR IN</sub>		tbd	tbd	mA
Power Supply Rejection Ratio	PSRR			0.5	% / % $\Delta$ V <sub>AA</sub>
(COMP = 0.1 $\mu$ F, f = 1 kHz)					

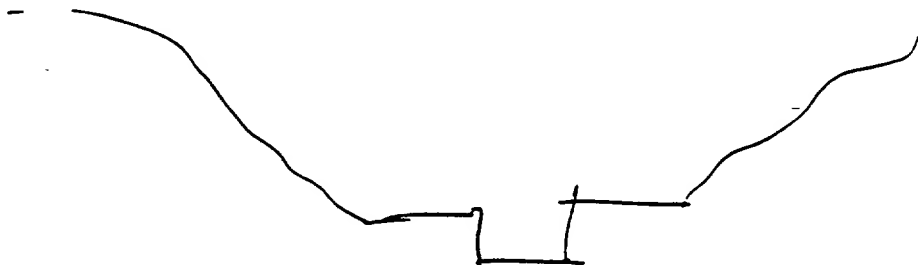
Test conditions to generate RS-343A standard video signals (unless otherwise specified): "Recommended Operating Conditions" using external voltage reference with SETUP = 7.5 IRE, RSET = 147  $\Omega$ , VREF = 1.235 V. As the above parameters are guaranteed over the full temperature range, temperature coefficients are not specified or required.

When the internal voltage reference is used, RSET may require adjustment to meet these limits. Also, the "gray-scale" output current (white level relative to black) will have a typical tolerance of  $\pm 10\%$  rather than the  $\pm 5\%$  specified above.

When the device is in the 6-bit mode, the output levels are approximately 1.5% lower than these values.

Note 1: Onboard VREF numbers subject to change upon completion of characterization.

7.5 IRE  
Sync  
C



## AC Characteristics

Parameter	Symbol	110 MHz Devices			Units
		Min	Typ	Max	
OSC, OSC* All Mux Rates	Fmax			110	MHz
RS0-RS3 Setup Time <i>WA 1 RD</i>	1	10			ns
RS0-RS3 Hold Time	2	10			ns
RD* Asserted to D0-D7 Driven <i>assert time</i>	3	2			ns
RD* Asserted to D0-D7 Valid <i>valid time</i>	4			40	ns
RD* Negated to D0-D7 3-States <i>disables</i>	5			20	ns
Read D0-D7 Hold Time <i>RD</i>	6	2			ns
Write D0-D7 Setup Time <i>WA</i>	7	10			ns
Write D0-D7 Hold Time	8	10			ns
RD*, WR* Pulse Width Low	9	50			ns
RD*, WR* Pulse Width High <i>2 RD</i>	10	6* pixel clock periods			ns
GLCLK Rates	Gmax				
8:1 Multiplexing				13.75	MHz
4:1 Multiplexing				27.5	MHz
2:1 Multiplexing				55	MHz
1:1 Multiplexing				90	MHz
VLCLK Rate	Vmax			85	MHz
DIVCLK1, DIVCLK2 Rates	Dmax			55	MHz
OSC, OSC* Cycle Time (Note 1)	11	18.18			ns
All Mux Rates					
OSC, OSC* Pulse Width High	12	tbd			ns
All Mux Rates					
OSC, OSC* Pulse Width Low	13	tbd			ns
All Mux Rates					
Duty Cycle of Selected Pixel Clock		45		55	%
When Clock Doubler Enabled					
GLCLK Cycle Time	14				
8:1 Multiplexing		72.72			ns
4:1 Multiplexing		36.36			ns
2:1 Multiplexing		18.18			ns
1:1 Multiplexing		11.11			ns
GLCLK Pulse Width High	15				
8:1 Multiplexing		4			ns
4:1 Multiplexing		4			ns
2:1 Multiplexing		4			ns
1:1 Multiplexing		4			ns
GLCLK Pulse Width Low	16				
8:1 Multiplexing		4			ns
4:1 Multiplexing		4			ns
2:1 Multiplexing		4			ns
1:1 Multiplexing		4			ns

Test conditions at end of this section

P1 f. line

		110 MHz Devices			
Parameter	Symbol	Min	Typ	Max	Units
VLCLK Cycle Time <i>n/c</i>	17	11.76			ns
VLCLK Pulse Width High	18	4			ns
VLCLK Pulse Width Low	19	4			ns
DIVCLK1, DIVCLK2 Cycle Time	20	14.81			ns
DIVCLK1, DIVCLK2 Duty Cycle	21	40		60	%
Graphics Data Setup to GLCLK	22	3			ns
Graphics Data Hold from GLCLK	23	1			ns
Data Setup to GLCLK ENABLE, BLANK*, C/HSYNC*, VSYNC*	24	3			ns
Data Hold to GLCLK ENABLE, BLANK*, C/HSYNC*, VSYNC*	25	1			ns
Video Data Setup to VLCLK	26	3			ns
Video Data Hold from VLCLK	27	1			ns
VALID Setup to VLCLK	28	3			ns
VALID Hold from VLCLK	29	1			ns
VLCLK to READY Valid	30			7	ns
DIVCLK1 to SEN Valid	31			3	ns
FIFO Reset Pulse Width		2 * VLCLK periods			ns
Analog Output Delay	32			30	ns
Analog Output Rise/Fall Time	33		3		ns
Analog Output Settling Time (Note 2)	34		13		ns
Clock and Data Feedthrough (Note 2)			-30		dB
Glitch Impulse (Note 2)			75		pV - sec
SENSE* Output Delay	35		1		μs
DAC-to-DAC Crosstalk			-23		dB
Analog Output Skew				2	ns
VAA Supply Current Normal Operation	IAA		tbd	tbd	mA
"Sleep" Mode (Note 3)			tbd	tbd	mA

**Test conditions at end of this section**

## AC Characteristics (continued)

Pipeline Delay	
Graphics 1:1/No Video	3 LCLKS + 16 PCLKS
MUX Graphics/No Video	(8 LCLKS + 16 PCLKS) ± 2 LCLKS
Graphics 1:1/Video	27 LCLKS + 16 PCLKS
MUX Graphics/Video	(32 LCLKS + 16 PCLKS) ± 2 LCLKS
The number of LCLKS will have to be multiplied by the respective MUX rates to get the proper number of pipeline delays. (i.e., PCLK = Pixel Clock Rate LCLK = MUX Clock Rate the pipeline delay in 2:1 MUX Graphics/No video, measured in PCLKs = 32 PCLKS ± 4 PCLKs.	

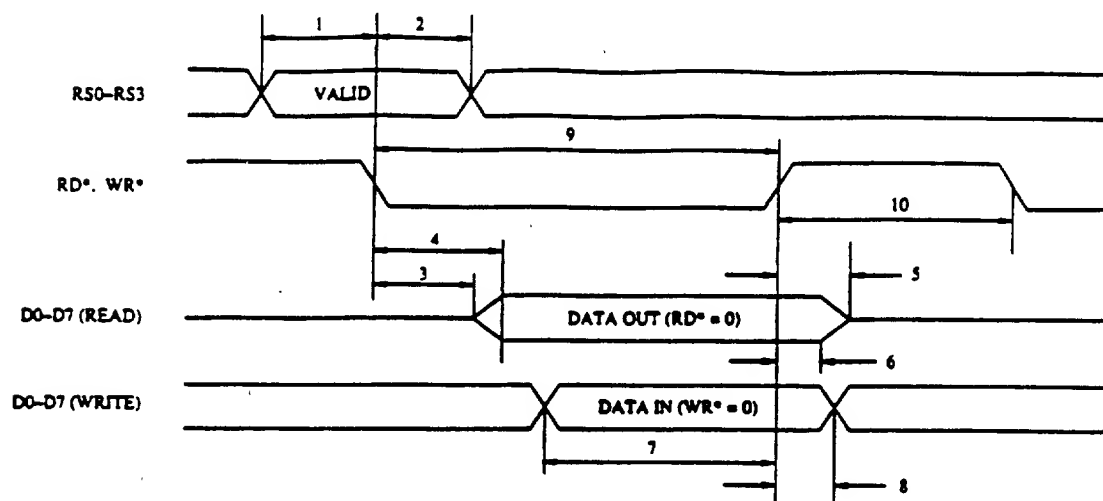
Test conditions (unless otherwise specified): "Recommended Operating Conditions" using external voltage reference with SETUP = 7.5 IRE, VREF = 1.235 V, RSET = 147Ω. TTL input values are 0-3 V, with input rise/fall times ≤ 3 ns, measured between the 10% and 90% points. Timing reference points at 50% for inputs and outputs. Analog output load ≤ 10 pF; SENSE\* and D0-D7 output load ≤ 50 pF. DIVCLK1, DIVCLK2 output load = 50 pF. As the above parameters are guaranteed over the full temperature range, temperature coefficients are not specified or required. Timing waveforms are shown in Figure 9 through Figure 11.

Note 1: OSC and OSC\* cycle times assume the use of the 2x Clock Multiplier.

Note 2: Numbers guaranteed by design.

Note 3: External voltage reference is disabled during sleep mode. all inputs are low, and clock is running.

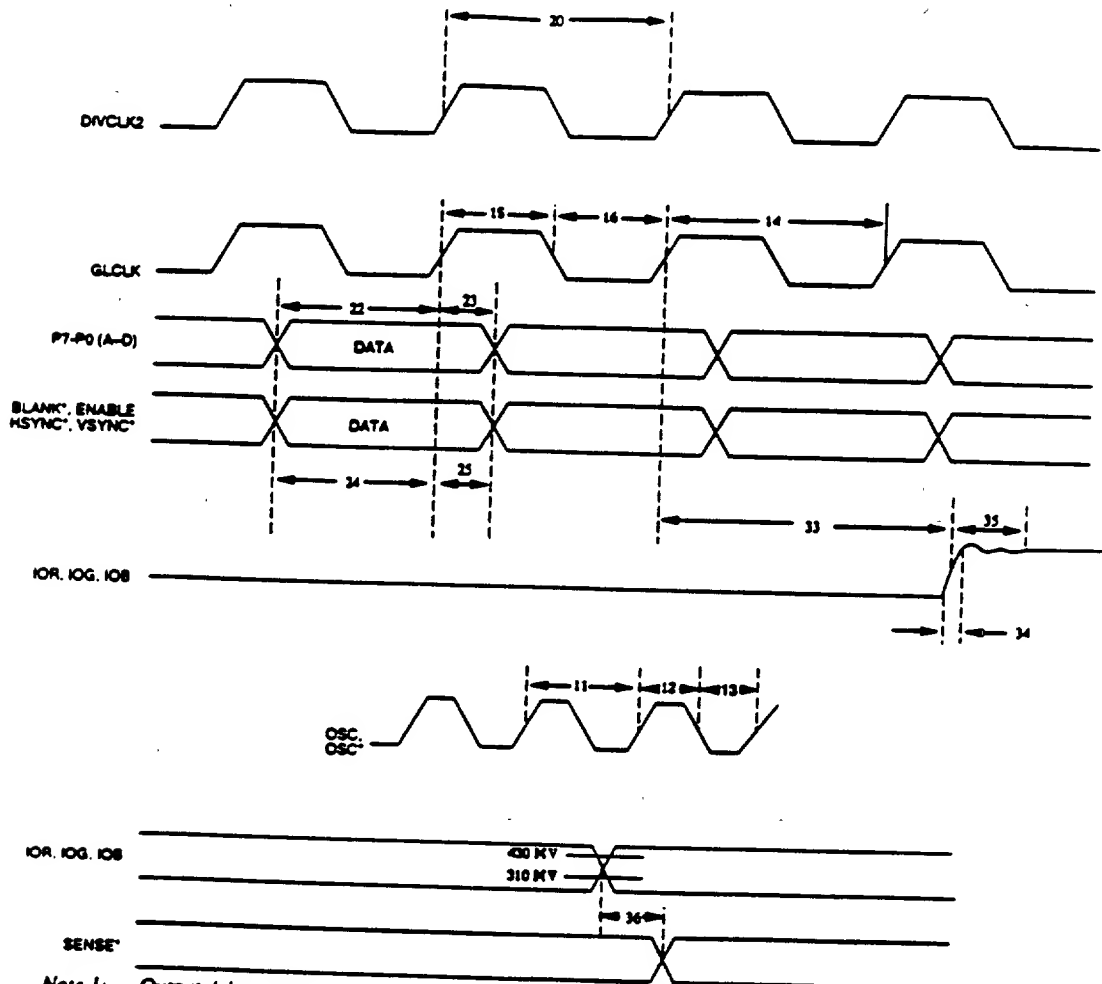




- Note 1:** Output delay measured from the 50% point of the rising edge of CLOCK to the 50% point of full-scale transition.  
**Note 2:** Settling time measured from the 50% point of full-scale transition to the output remaining within  $\pm 1$  LSB.  
**Note 3:** Output rise/fall time measured between the 10% and 90% points of full-scale transition.

**Figure 9. MPU Read/Write Timing.**

## Timing Waveforms (continued)



- Note 1:** Output delay measured from the 50% point of the rising edge of CLOCK to the 50% point of full-scale transition.
- Note 2:** Settling time measured from the 50% point of full-scale transition to the output remaining within  $\pm 1$  LSB.
- Note 3:** Output rise/fall time measured between the 10% and 90% points of full-scale transition.

Figure 10. Graphics Input/Output Timing.

Timing Waveforms (continued)

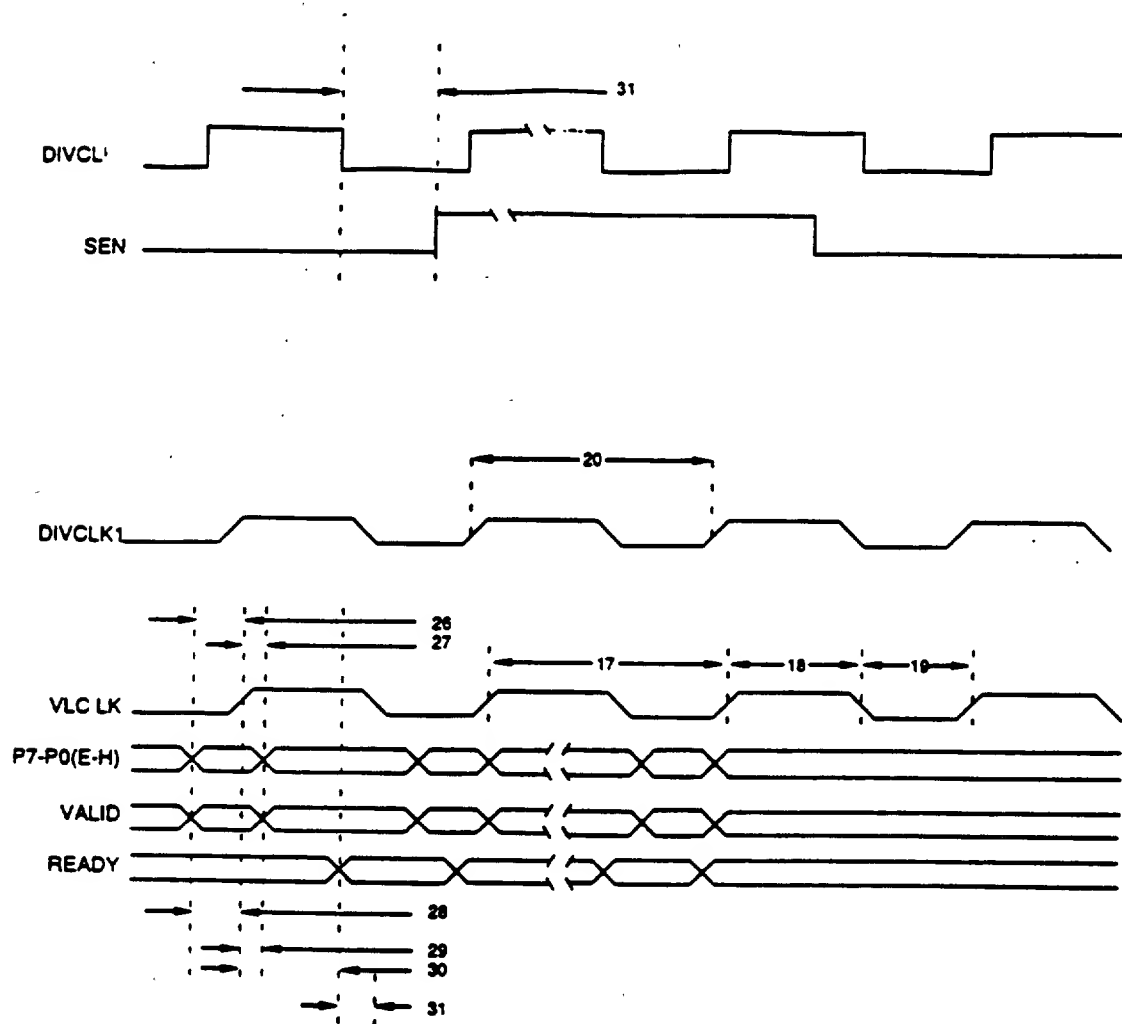


Figure 11. Video Input/Output Timing.

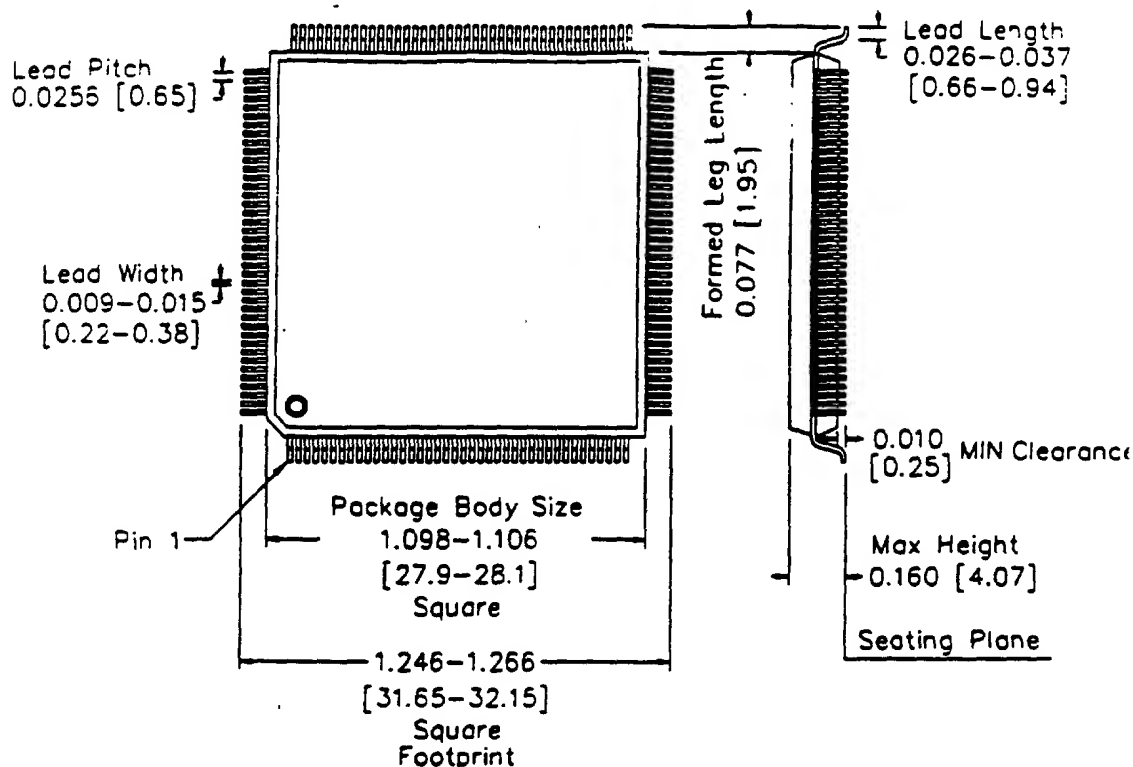
## Ordering Information

Model Number	Speed	Package	Ambient Temperature Range
Bt885KHF110	110 MHz	160-pin Plastic Quad Flatpack	0°C to + 70°C

## Revision History

Datasheet Revision	Changes From Previous Revision
B	Initial Release
C	<p>Pinout change. Pin 160 was changed from GND to RPTLINE, Pin 2 was changed from GND to VRESET*. Pins 1, 39, 40, 41, 42, 79, 80, 81, 82, 119, 120, 121, 122 were changed from GND to NC.</p> <p>Vertical scaling support added. Both DIVCLK1 and DIVCLK2 and all internal clocks are derived from OSC or OSC* clock inputs.</p> <p>DIVCLK1 and DIVCLK2 outputs are opposite phases.</p>
D	<p>135 MHz speed grade removed.</p> <p>Pinout change. Pin 160 was changed from RPTLINE to NC.</p> <p>Vertical scaling deleted.</p>

Package Drawing-160-pin Plastic Quad Flatpac (PQFP)



Notes. Unless otherwise specified:

1. Dimensions are in inches [millimeters]. Millimeters are the controlling dimension.
2. Package body size does not include mold protrusion or mismatch.
3. PCB pad layout suggestions:
  - a. Pad size: 0.100 x 0.012 [2.54 x 0.30].
  - b. Lead pitch (millimeters): Use 0.65 center-to-center spacing.
  - c. Lead pitch (inches): If the PCB layout system to be used can handle fractional mils, use 0.0256 center-to-center spacing. If not, use a combination of 0.025(A) and 0.026(B) inch spacings in groups of five ("ABABA" repeated) to approximate the exact spacing as closely as possible. For example, "ABABA" "ABABA" and so forth.

# Brooktree®

**Brooktree Corporation**  
9868 Scranton Road  
San Diego, CA 92121-3707  
(619) 452-7580  
1(800) 2-BT-APPS  
TLX: 383 596  
FAX: (619) 452-1249  
Internet: [apps@brooktree.com](mailto:apps@brooktree.com)  
L885001 Rev. D

### CAUTION



**ESD-sensitive devices.**  
Permanent damage may occur on unconnected devices subjected to high-energy electrostatic fields. Unused devices must be stored in conductive foam or shunts.  
Do not insert this device into powered sockets.  
Remove power before insertion or removal.



printed on recycled paper

AT1019958

## Circuit Description

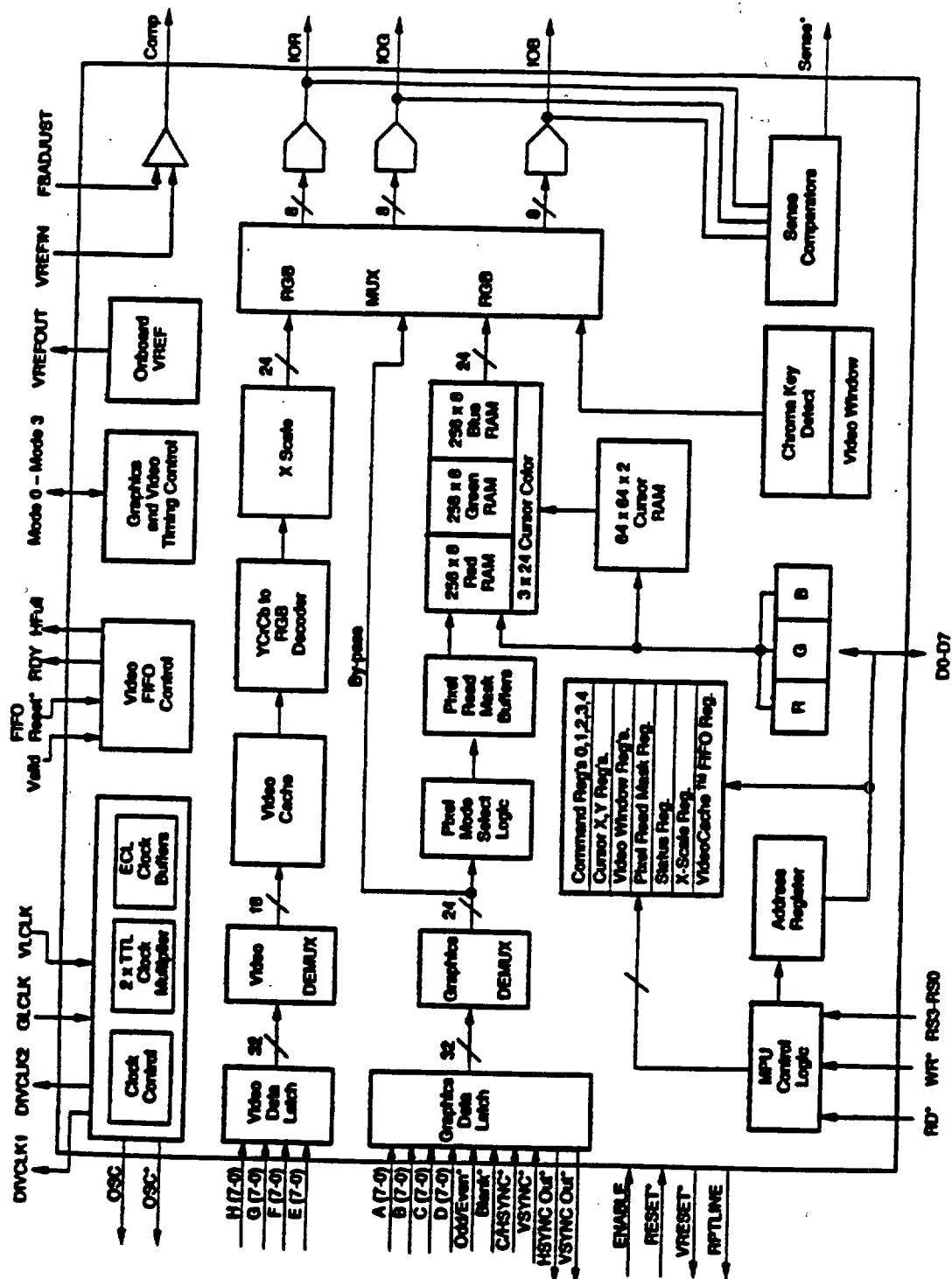


Figure 1. Bt885 Detailed Block Diagram.

EXHIBIT

RX-240

### Advance Information

This document contains information on a product under development. The parametric information contains target parameters that are subject to change.

### Distinguishing Features

- 110 MHz Pipelined Operation
- VGA Compatible
- Mixed Video and Graphics
- 32-bit Graphics and 32-bit Video Pixel Ports
- YCrCb-to-RGB Conversion
- YCrCb 4:2:2 and 2:1:1 Interpolation
- Uses Brooktree's VideoCache™ Technology
- Horizontal Video Up-Scaling
- 64 x 64 x 2 Cursor
- VRAM Shift Clock Support
- Enables DRAM-Based Motion Video Systems
- Programmable Video Extents
- Programmable Color Keying
- Onboard TTL Clock Doubler
- Three 256 x 8 Color Palette RAMs
- Simplifies Integration of Video into Microsoft Windows™
- 3 x 24 Cursor Color Palette
- Standard MPU Interface
- Power-Down Mode
- Directly Implements Brooktree's VideoCache™ Connector
- 160-Pin PQFP Package

### Applications

- Video Decompression Acceleration
- Multimedia Workstations
- High-Resolution Graphics
- Desktop Video

### Related Products

- Bt812 Video Decoder
- Bt858 Video Encoder
- Bt895 Video Controller
- Bt81295 Personal Media Adapter

# Bt885

## 110 MHz Monolithic CMOS Video CacheDAC™

### Product Description

The Bt885 is designed specifically for dual or unified frame buffer multimedia subsystems. A dedicated video port accepts a CCIR601 YCrCb or RGB data stream and allows on-screen switching on a pixel-by-pixel basis. Mixing occurs within programmable video extents based on a flexible color key mechanism. Bt885 is intended to replace multiple RAMDAC™-based multimedia subsystems. The Bt885 register set is VGA compatible.

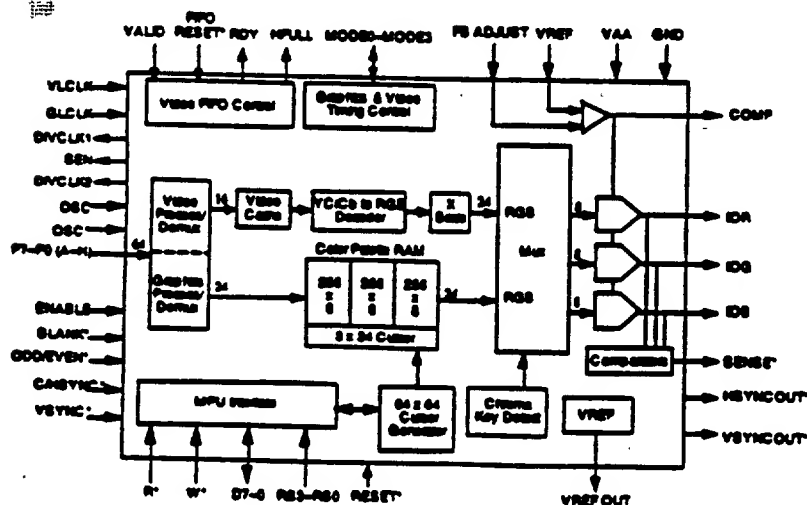
The Bt885 can accelerate video decompression and work with the Bt812 decoder chip using programmable interpolation to pixel multiply by 1, 2, or 4 for CCIR601 4:2:2, 2:1:1, and 1:0.5:0.5 formats. This allows the video data to mix with the graphics data at the same rate.

Brooktree's 800-byte VideoCache™ FIFO enables asynchronous delivery of graphics and video, easing system bandwidth requirements for video transfer, and allowing efficient use of system memory. Non-integer scaling permits arbitrary video window sizing.

The 64 x 64 x 2 bit cursor has its own palette and has priority over the video or graphics. The cursor operates in three modes: Microsoft Windows™, three-color, and X Windows.

The Bt885 supports independent 32-bit graphics and 32-bit video pixel ports and is compatible with both VRAM- and DRAM-based video subsystems.

The Bt885 generates RS-343A-compatible video signals into a doubly terminated 75 Ω load.



Brooktree Corporation • 9868 Scranton Road • San Diego, CA 92121  
 (619) 452-7580 • (800) 2BT-APPS • TLX: 383 596 • FAX: (619) 452-1249  
 L885001 Rev. D

Brooktree®

AT1019959



## Circuit Description (continued)

**MPU Interface**

As illustrated in the detailed block diagram (Figure 1), a standard MPU bus interface is supported, allowing the MPU direct access to the color palette RAM. MPU data is transferred into and out of the CacheDAC™ through the D0-D7 data pins. The read/write timing is controlled by the RD\* and WR\* inputs.

The RS0-RS3 select inputs specify which control register the MPU is accessing, as shown in Table 1. The 8-bit address register is used to address the color palette RAM, eliminating the requirement for external address multiplexers. D0 corresponds to ADDR0 and is the least significant bit.

**Hardware Reset Condition**

On reset, Bt885 is configured for standard VGA compatibility as follows:

- 8 bits per pixel graphics, 1:1 MUX.
- 6-bit DAC resolution.
- Pixel mask register set to 0xFF.
- Video modes disabled.
- All control registers set for VGA compatibility.
- Graphic pipelines are reset.

**Writing Color Palette RAM Data**

To write color data, the MPU writes the address register (RAM write mode) with the address of the color palette RAM location to be modified. The MPU performs three successive write cycles (8 bits each of red, green, and blue), using RS0-RS3 to select the color palette RAM. After the blue write cycle, the 3 bytes of color information are concatenated into a 24-bit word and written to the location specified by the address register. The address register then increments to the next location, which the MPU may modify by writing another sequence of red, green, and blue data. A block of color values in consecutive locations may be written to by writing the start address and performing continuous RGB write cycles until the entire block has been written. Refer to the Timing Waveforms section for further information.

**Reading Color Palette RAM Data**

To read color palette RAM data, the MPU loads the address register (RAM read mode) with the address of the color palette RAM location to be read. The contents of the color palette RAM at the specified address are

copied into the red, green, or blue (RGB) registers and the address register is incremented to the next RAM location. The MPU performs three successive read cycles (8 bits each of red, green, and blue), using RS0-RS3 to select the color palette RAM. Following the blue read cycle, the contents of the color palette RAM at the address specified by the address register are copied into the RGB registers, and the address register increments again. A block of color values in consecutive locations may be read by writing the start address and performing continuous RGB read cycles until the entire block has been read.

RS3-RS0	Access	Addressed by MPU
0000	R/W	address register; palette/cursor RAM write
0001	R/W	6/8-bit color palette data
0010	R/W	pixel mask register
0011	R/W	address register; palette/cursor RAM read
0100	R/W	address register; cursor/overscan color write
0101	R/W	cursor overscan and color data
0110	R/W	command register 0
0111	R/W	address register; cursor/overscan color read
1000	R/W	command register 1
1001	R/W	command register 2
1010	R/W	extended address read/write register
1011	R/W	cursor RAM array data
1100	R/W	cursor x-low register
1101	R/W	cursor x-high register
1110	R/W	cursor y-low register
1111	R/W	cursor y-high register

**Table 1. Control Input Truth Table**  
(RS3 = MSB, RS0 = LSB).

**Writing Cursor and Overscan Color Data**

To write cursor or overscan color data, the MPU writes the address register (cursor color write mode) with the address of the cursor or overscan color location to be modified. The MPU performs three successive write cycles (8 bits each of red, green, and blue), using RS0-RS3 to select the cursor color registers. After the blue write cycle, the 3 bytes of red, green, and blue color information are concatenated into a 24-bit word and written to the cursor or overscan color location speci-

## Circuit Description (continued)

fed by the address register. The address register then increments to the next location, which the MPU may modify by writing another sequence of red, green, and blue data. A block of color values in consecutive locations may be written to by writing the start address and performing continuous RGB write cycles until the entire block has been written.

### Reading Cursor Color Data

To read cursor color data, the MPU loads the address register (cursor color read mode) with the address of the cursor color location to be read. The contents of the cursor color register at the specified address are copied into the RGB registers, and the address register is incremented to the next cursor color location. The MPU performs three successive read cycles (8 bits each of red, green, and blue), using RS0–RS3 to select the cursor color registers. Following the blue read cycle, the contents of the cursor color location at the address specified by the address register are copied into the RGB registers, and the address register again increments. A block of color values in consecutive locations may be read by writing the start address and performing continuous R, G, B read cycles until the entire block has been read.

### Extended Register Mechanism

An extended register set is used to accommodate all features of the Bt885. Since there are only four register select lines (and all 16 combinations have already been used), the extended registers must be accessed indirectly.

For example, Command Register 3 is accessed with the following sequence of operations:

1. Set RS3–RS0 = 0000, Address Register.
2. Write Address Register to 0x02
3. Set RS3–RS0 = 1010 (Extended Address Register).
4. Read or Write Command Register 3.

### Writing Color Key Color Data

To write the color key color data value, the MPU selects the color key data RGB register using the extended register. It then performs a write cycle setting RS3–RS0 to 1010 (Status Register). This process is repeated for each color component. The color key color register is only updated after the blue value is written.

### Reading Color Key Color Data

To read the color key color data value, the MPU selects the color key data RGB register using the extended register mechanism, then performs a read cycle setting RS3–RS0 to 1010 (Status Register).

### Writing Color Key Mask Data

To write the color key mask data value, the MPU selects the color key mask RGB register using the extended register mechanism. It then performs a write cycle setting RS3–RS0 to 1010 (Status Register). This process is repeated for each color component. The color key mask register is only updated after the blue value is written.

### Reading Color Key Mask Data

To read the color key color mask value, the MPU selects the color key data RGB register using the extended register mechanism outlined below, then performs a read cycle setting RS3–RS0 to 1010 (Status Register).

### Additional Information

When the color palette RAM is accessed, the address register resets to 0x00 following a blue read or write cycle to RAM location 0xFF.

The MPU interface operates asynchronously to the pixel clock. Data transfers between the color palette RAM and the color registers (R, G, and B in the functional block diagram) are synchronized by internal logic, and occur in the period between MPU accesses. To reduce noticeable sparkling on the CRT screen during MPU access to the color palette RAMs, internal logic maintains the previous output color data on the analog outputs while the transfer between RGB registers and lookup table RAMs occurs.

To keep track of the red, green, and blue read/write cycles, the address register has two additional bits (ADDRa and ADDRb) that count modulo three. They are reset to zero when the MPU writes to the address register and are not reset to zero when the MPU reads the address register. The MPU does not have write access to these bits. The MPU may read the address register at any time without modifying its contents or the existing read/write mode. These bits can be read from SR1.

## Circuit Description (continued)

### Accessing the Cursor RAM Array

The 64 x 64 x 2 cursor RAM is accessed in a planar format. Bits CR31 and CR30 in Command Register 3 become the load inputs to the 2 MSBs of a 10-bit address counter; therefore, these bits must be written in Command Register 3 before the lower 8 bits are written to the address counter through the MPU port. In the planar format, only nine address bits are used. The tenth bit determines which plane (0 or 1) data of the cursor RAM array is accessed. A single address presented to the cursor RAM accesses 8-bit locations in plane 0 or 1, depending on the state of address bit 9.

After each access in the planar format, the address increments. The MPU uses ADDR, a 10-bit binary address counter, to access the cursor RAM array. The address counter is the same 8-bit binary counter used for RGB autoincrementing with CR31 and CR30 as its extended MSBs. Any write to the address counter after cursor autoincrementing has been initiated resets the cursor autoincrementing logic until cursor RAM array has again been accessed. Cursor autoincrementing will then begin from the address written. A read from the address counter does not reset the cursor, autoincrementing logic. The color palette RAM and the cursor RAM share the same external address register, and MPU addressing for this and all other registers is determined by the external register select lines RS3-RS0 (see Table 2).

### 6-Bit/8-Bit Operation

The command bit CR01 specifies whether the MPU is reading and writing 8 bits or 6 bits of color information each cycle. For an 8-bit operation, D0 is the LSB and D7 is the MSB of color data. For a 6-bit operation, color data is contained on the lower 6 bits of the data bus, with D0 as the LSB and D5 as the MSB of color data. When the MPU is writing color data, D6 and D7 are ignored. During color read cycles, D6 and D7 are a logical zero.

Accessing the cursor RAM array does not depend on the resolution of the DACs. When Bt885 is in the 6-bit mode, the 6-bit DAC values are left justified within an 8-bit field and the two LSBs are set to zero. Therefore, Bt885's full-scale output current will be about 1.5% lower than while it is in the 8-bit mode.

### Power-Down Mode

The Bt885 incorporates a power-down capability, controlled by command bit CR00. While command bit CR00 is a logical zero, the Bt885 functions normally.

While command bit CR00 is a logical one, the DACs, cursor circuitry, video FIFO, and power to the RAM are turned off. The RAM still retains the data. Also, the RAM may be read or written to by the MPU as long as the pixel clock is running. The RAM automatically powers up during MPU read/write cycles and shuts down when the MPU access is completed. The DACs output no current, and the four command registers may still be written to or read by the MPU. The output DACs require about one second to turn off (sleep mode) or turn on (normal), depending on the compensation capacitor used (see the Video Output Waveforms section for further information). The DACs will be turned off during sleep mode only if a voltage reference (internal or external) is used.

When an external voltage reference is used, external circuitry should turn off the voltage reference ( $V_{REF} = 0$  V) to further reduce power consumption caused by biasing of portions of the internal voltage reference.

### Pixel Clock Selection

OSC and OSC\* provide the source for the Bt885 internal pixel clock. Graphic pixel data is latched by GLCLK. Bit CR24 selects whether the OSC or OSC\* pin is used. A clock doubler can be enabled on the selected input by setting CR33 = 1. The OSC\* and OSC inputs can be used together as differential ECL inputs for the external clock by setting CR34 = 1. If a differential ECL input mode is used (CR34 = 1), then the state of CR24 is ignored. The state of CR33 must be 0.

It is also possible to internally route the DIVCLK2 output to the latches connected to GLCLK by setting CR36 = 1. GLCLK will be ignored in this mode.

DIVCLK1 and DIVCLK2 are output on the basis of the OSC and OSC\* inputs as described unless they are disabled by setting CR32 = 0 (DIVCLK1 disable) or CR35 = 0 (DIVCLK2 disable). If the clock doubler is used (CR33 = 1), then both the DIVCLK1 and DIVCLK2 dividers must be set to a value of 2 or greater. DIVCLK1 and DIVCLK2 are opposite phases.

## Circuit Description (continued)

CR31 (bit A8 of ADDR)	ADDR 0-7 (counts binary)	ADDR a,b (counts modulo 3)	RS3	RS2	RS1	RS0	Addressed by MPU
N/A	0x00-0xFF	00 01 10	0 0 0	0 0 0	0 0 0	1 1 1	Color palette RAM (Red Component) Color palette RAM (Green Component) Color palette RAM (Blue Component)
N/A	0x00	00 01 10	0 0 0	1 1 1	0 0 0	1 1 1	Overscan color (Red Component) Overscan color (Green Component) Overscan color (Blue Component)
N/A	0x01	00 01 10	0 0 0	1 1 1	0 0 0	1 1 1	Cursor Color 1 Red Component Cursor Color 1 Green Component Cursor Color 1 Blue Component
N/A	0x02	00 01 10	0 0 0	1 1 1	0 0 0	1 1 1	Cursor Color 2 Red Component Cursor Color 2 Green Component Cursor Color 2 Blue Component
N/A	0x03	00 01 10	0 0 0	1 1 1	0 0 0	1 1 1	Cursor Color 3 Red Component Cursor Color 3 Green Component Cursor Color 3 Blue Component
0 1	0x000-0x1FF 0x200-0x3FF	N/A N/A	1 1	0 0	1 1	1 1	Cursor RAM Array, plane 0 Cursor RAM Array, plane 1

Table 2. Address Register Operation and Autoincrementing.

## Frame Buffer Pixel Port Interface

There are 64 input pins P0-P7 (A-H) used to interface to the graphics and video frame buffer memories. The assignment of pins to input pixels is determined by the operation mode and multiplex rate.

## Video Port Clocking

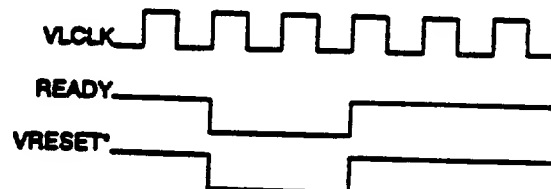
Video data is synchronously clocked into Bt885 with the VLCLK input. VLCLK may be asynchronous from the pixel and/or graphics load clock, as an internal FIFO is used to synchronize video data to graphics pixel data.

Three status signals are available to control the loading of video pixel data into Bt885: VALID, READY and HFULL. VALID is provided by the system to Bt885 and is asserted to indicate that valid video data is being presented on the video pixel port. The READY signal is an output from Bt885 that indicates that it is accepting pixel data. For data to be accepted on any particular VLCLK rising edge, both the VALID and READY signals must be high through the clock edge. The HFULL signal is used to keep check on how full the video cache is and helps to prevent overloading the internal video FIFO.

The system must load video data into Bt885 prior to the time that it is to be used. In systems where there is a

one-to-one relationship between video pixels and graphics pixels in the frame buffer and this data is delivered simultaneously, the FIFO operation can be ignored and VALID would be tied to the pixel blanking signal from the graphics subsystem (BLANK\*). In this mode, the FIFO would never be filled and, therefore, READY may be ignored.

The internal video data FIFO is reset to an empty state on each detected vertical blank period. The system can immediately begin loading data into the video port regardless of the video window's position on the screen. With the VRESET\* signal (see pin description) the video data will know the start of each graphics frame. See diagram below:



If at any time the video FIFO is empty when video data is required, Status Register 2 bit SR27 will be set to one. The underflow bit will remain set until Status Register 2 is written, then SR27 will be cleared.

For proper operation of the video pipeline reset, VLCLK must be a free-running clock.

## Circuit Description (continued)

**VideoCache™ FIFO Operation**

The Bt885 provides a FIFO buffer for video pixels to allow for asynchronous video and graphics operation, and to ease system design requirements. Use of the VideoCache™ FIFO features is entirely optional and not necessary for synchronous designs.

**Loading VideoCache™ FIFO**

The VideoCache™ FIFO accepts a group of data (the exact number is given by the current video mode) when the following conditions are met on any single rising edge of VLCLK:

1. The FIFO is ready to accept data (i.e., it is not full). This is determined by the state of the READY signal.
2. The system is presenting data, indicating this to the CacheDAC™ by asserting the VALID signal with the data.

**Unloading VideoCache™ FIFO**

Bt885 will unload the VideoCache™ FIFO dependant on the setting of bit CR41. If CR41 = 1, the video will only be unloaded while Bt885 is scanning through the video window. If CR41 = 0, then video will always be unloaded during active graphics time. The unloading process is independent of color keying.

**MFULL**

This signal is asserted when the VideoCache™ FIFO is more than half full.

**VideoCache™ Reset**

There are four ways that the VideoCache FIFO gets reset:

**FIFO Reset Pin.** This is an external hardware FIFO RESET method for resetting the Bt885 Video FIFO. This pin must be held low for at least two VLCLKs with PDCEL CLOCK running.

**CR47.** This is a software RESET method for resetting the Bt885 VideoCache FIFO. A logical one written to this bit resets the VideoCache FIFO after four VLCLKs. A logical zero will put the FIFO back to normal operation.

**An under flow occurred.** The SR27 status bit says a VIDEO FIFO under flow occurred when a logical one is read. The VideoCache FIFO is automatically reset when this happens. A MPU write cycle to Status Register-2 will clear SR27.

**Vertical Retrace Interval.** An automatic VIDEO FIFO reset also occurs during the vertical retrace interval. When the ENABLE line is low for 2048 clock cycles an internal FIFO RESET sequence is initiated.

READY and VRESET\* will be = 0 during the FIFO reset period. READY will become active within one PDCEL CLOCK period after VRESET\* goes high.

**General Purpose Signals****DIVCLK1 / DIVCLK2**

These signals provide programmable free-running clocks based on the internal pixel clock. They can be used to generate external pixel load clocks, such as VLCLK or GLCLK. A gated clock may be generated from DIVCLK1 by using another general purpose signal, SEN, described below.

**SEN**

This signal is used to provide a gating control for DIVCLK1. SEN can be programmed to start relative to the falling edge of internally detected vertical blank (see cursor operation) in units of scanlines and relative to the falling edge of C/HSYNC\* in DIVCLK1 cycles using the serial clock enable start (horizontal and vertical) registers. Duration is set in units of scanlines for the vertical direction and in DIVCLK1 cycles for the horizontal direction (relative to the beginning of SEN) using the serial clock enable duration (horizontal and vertical) registers. This signal is guaranteed to transition only during DIVCLK1 low time.

This signal may be used to control a VRAM shift clock which runs during non-blanking time. When an appropriate delay is programmed from the leading edge of C/HSYNC\*, the serial data can be properly positioned before the trailing edge of BLANK\*. The SEN duration register then stops the serial clock to allow the system to perform VRAM row data transfer. Because C/HSYNC\* is sampled with the internal pixel clock, there may be an additional pixel clock delay between C/HSYNC\* falling and the SEN rising.

**MODE0**

This is a general purpose, TTL compatible, registered input/output which is set using CR45. Selection of input or output is made using CR37.

This pin is user-definable and could be used to interface between hardware and software. For example, MODE0 could detect the existence of a video card. Software would detect this by reading CR45.

## Circuit Description (continued)

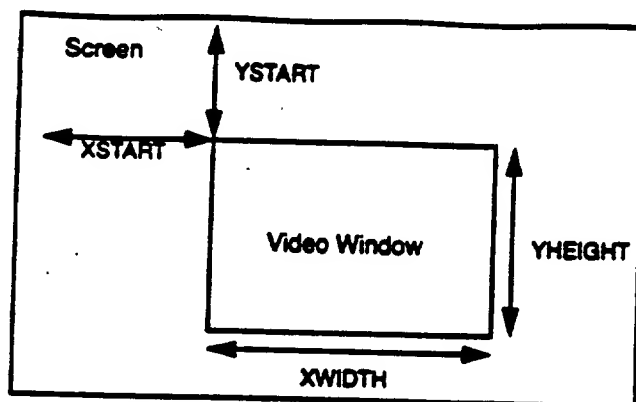


Figure 2. Video Window Registers.

## Video Window Operation

The XSTART register indicates the starting X position on the screen for the video window relative to the ENABLE pin (Figure 2). A value of zero indicates that the video window begins with the first (leftmost) pixel of each horizontal scan line. The YSTART register indicates the starting Y position on the screen for the video window. A value of zero indicates that the video window begins on the first scan line of each frame. The XWIDTH register indicates the number of pixels per scan line within the video window. A value of zero indicates that there are no pixels in the video window. The YHEIGHT register indicates the number of scan lines within the video window. A value of zero indicates that there are no scan lines in the video window.

All four values, XSTART, XWIDTH, YSTART, and YHEIGHT should be written sequentially. Internal video window coordinates are loaded during the next detected vertical blanking interval after the YHEIGHT register is written.

## Video Scaling Operation

The Bt885 supports video upscaling in the horizontal direction. Horizontally, a combination of coarse pixel interpolation and pixel-accurate replication may be applied. Downscaling of the source image, both horizontally and vertically, must be performed outside the Bt885.

## Horizontal Scaling

Horizontal upscaling may be accomplished by using a combination of two methods: pixel replication and pixel interpolation.

Pixel replication is accomplished by using the output of an overflowing 12-bit accumulator to either clock a value out of the VideoCache™ FIFO to the DACs or to hold the current DAC value.

At the start of each scan line, the accumulator is initialized to the value stored in the XSCALEINIT register. On each pixel, the value stored in the XSCALEINC register is added to the accumulator.

If the addition results in a carry, a pixel is clocked out of the VideoCache™ FIFO to the DACs. If no carry occurs, the previous DAC value is held. This style of scaling is known as a Digital Differential Algorithm (DDA).

To accomplish scaling, the system supplying the Bt885 with video pixel must precalculate the DDA constants required for the desired scale factor and load the values into the two 12-bit X-scaling registers, XSCALEINIT and XSCALEINC, as follows:

$$\text{XSCALEINC} = \left[ \frac{(\text{Source Video Width} \cdot 0x1000) + 0x0800 - \text{XSCALEINIT}}{\text{Destination Video Width}} \right]$$

$0 \leq \text{XSCALEINIT} \leq 0x0FFF$ . XSCALEINIT can be used to set the replication phase of the DDA in more advanced applications.

## Circuit Description (continued)

Pixel interpolation is available when using certain YCrCb video modes. When used, it can interpolate the data to 2x or 4x the source horizontal pixel count. The table below shows source data formats, video mode selected (CR1), and the resulting interpolation factor achieved.

Source Video Format	Video Mode Selected	Interpolation Factor Achieved
YCrCb 4:2:2	YCrCb 4:2:2	1:1
	YCrCb 2:1:1	2:1
	YCrCb 1:1:1	4:1
YCrCb 2:1:1	YCrCb 2:1:1	1:1
	YCrCb 1:1:1	2:1
YCrCb 1:1:1	YCrCb 1:1:1	1:1

**Example:** To fill a window which is 636 pixels wide with a source of 320 pixels of YCrCb 4:2:2 data loaded in 1:1 mux mode, one should select the YCrCb 2:1:1 video mode (CR17-CR14 = 5) and set XSCALEINC to 0x0FFF for no replication (see note below for why 636 pixels was chosen). If the window were slightly larger, say 700 pixels wide, one should select the YCrCb 2:1:1 video mode and use the pixel replicator to stretch the 636 new pixels into 700 pixels (XSCALEINC = 0x0E89, XSCALEINTT = 0x0800).

## Color Key Operation

Selection between the video and graphics pixel data may be based on a specified range of graphic pixel values. A "color key set" may be defined which specifies one or more graphic pixel values that allow video pixels to be shown.

To define the color key set, three color key registers and three color mask registers are used. A graphic pixel value is bitwise XORed with the color key and the result is NANDed with the color mask. If the result is one, the corresponding video pixel is displayed in its place.

When a graphic pixel value falls within the color key set, the corresponding video pixel is displayed rather than the graphic pixel. Color key detection may occur either before the palette lookup or after the palette lookup. In 16- and 24-bit pixel modes, if palette bypass is enabled, selecting matching after the palette matches based on the actual values that would be applied to the DACs.

When matching after the palette, bit CR42 of Command Register 4 should be set to 1, and the color key registers and color mask registers represent 24-bit RGB values each. The registers are ordered with red at the lowest address, then green and blue.

When matching before the palette, bit CR42 of Command Register 4 should be set to zero. The color key registers and color mask registers represent unmultiplied graphic pixel values, with the red register as the least significant byte, then green and blue. Only the bits needed to represent the pixel are used. For example, an 8-bit pixel color key and mask use only the red registers, 16-bit pixels use only the red and green registers.

Pixel selection occurs only within the current video window boundaries, and only when bit CR46 of Command Register 4 is set to 0 to allow color key detection. When CR46 is set to 1, all pixels within the video window will display the video pixels, regardless of color mask and key register values.

The hardware cursor always has display priority over color key selection.

## Circuit Description (continued)

## Example 1

Match a specific 8-bit pseudo-color palette position (value 0xFE).

CR42 = 0 (matching before palette)

CR46 = 0 (allow color keying)

Color Mask: (B) 0xFF (G) 0xFF (R) 0xFF

Color Key: (B) 0xFF (G) 0xFF (R) 0xFE

## Example 2

Match a range of blue values between 0xC0 and 0xC7.

CR42 = 1 (matching after palette)

CR46 = 0 (allow color keying)

Color Mask: (B) 0xF8 (G) 0x00 (R) 0x00

Color Key: (B) 0xC0 (G) 0x00 (R) 0x00

## Example 3

Use bit 15 in a TARGA 15-bit true-color mode to perform color key.

CR43 = 0 (matching before palette)

CR46 = 0 (allow color keying)

Color Mask: (B) 0xFF (G) 0x80 (R) 0x00

Color Key: (B) 0xFF (G) 0x80 (R) 0x00

*Note:* To set the color key or color mask, all three indexes *must* be written, even if all three indexes are not used.

## YCrCb-to-RGB Matrix

The matrix converts the YCrCb video data to 24 bits of RGB data (8 bits each).

The YCrCb-to-RGB conversion is compliant with CCIR Recommendation 601-1 as follows:

$$R = 1.164(Y - 16) + 1.596(Cr - 128)$$

$$G = 1.164(Y - 16) - 0.813(Cr - 128) - 0.391(Cb - 128)$$

$$B = 1.164(Y - 16) + 2.018(Cb - 128)$$

## Modes of Operation—Graphics

## 4-Bits/Pixel Operation (8:1 MUX)

The 32 input bits are multiplexed 8:1 and configured for 4 bits/pixel. There are eight independent 4-bit pixel ports, P7:4 (A-D) and P3:0 (A-D). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every eight pixel clock cycles. The 4 bits from each port will select one of sixteen locations in the palette (see Table 11 in the Internal Registers section).

## 8-Bits/Pixel Operation (4:1 MUX)

The 32 input bits are multiplexed 4:1 and configured for 8 bits/pixel. There are four independent 8-bit pixel ports, (A-D). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every four pixel clock cycles. The 8 bits from each port will select 1 of 256 locations in the palette (see Table 11 in the Internal Registers section).

## 8-Bits/Pixel Operation (2:1 MUX)

The 16 input bits are multiplexed 2:1 and configured for 8 bits/pixel. There are two independent 8-bit pixel ports, (A-B). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every two pixel clock cycles. The 8 bits from each port will select 1 of 256 locations in the palette (see Table 11 in the Internal Registers section).

## 8-Bits/Pixel Operation (1:1 MUX)

The 8 input bits are multiplexed 1:1 and configured for 8 bits/pixel. There is one 8-bit pixel port, (A). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every pixel clock cycle. The 8 bits will select 1 of 256 locations in the palette (see Table 11 in the Internal Registers section).

## 16-Bits/Pixel Operation (2:1 MUX)

The 32 input bits are multiplexed 2:1 and configured for 16 bits/pixel. There are two independent 16-bit pixel ports, (A-B) and (C-D). The bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every two pixel clock cycles. The pixel bits multiplexed in this mode are from the same ports of RGB color formats of 5:5:5 or 5:6:5. P7D and P7B are ignored internally when the 5:5:5 color format is selected (see Table 11 in the Internal Registers section).



## Circuit Description (continued)

Bit CR24 in Command Register 2 can be programmed to enable or disable true-color palette bypass. When the bypass mode is selected, the pixel data bypasses the palette as well as the pixel mask, and is transferred to the proper MSBs of the respective DACs, the remaining LSBs are set to zeros. When the bypass mode is not selected, the pixel data indexes the palette, and color information is passed to the respective DACs. Bit CR22 in Command Register 2 determines whether palette addressing is sparse or contiguous. For sparse palette addressing, each independent color component of pixel data is mapped to the most significant bits of the respective palette address; the least significant bits are set to zero. For contiguous palette addressing, each independent color component of the pixel data is mapped to the least significant bits of the respective palette address; the most significant bits are set to zero. The color palette values indexed, for either sparse or contiguous addressing, are transferred to the DACs. When 5:5:5 or 5:6:5 color format is selected, the display can contain 32 K or 64 K simultaneous colors. The DACs can be configured for 6 or 8 bits of resolution in this mode. If 5:5:5 color format is selected, the most significant bit may be used for color key operation (see Tables 3 and 4).

## 16-Bits/Pixel Operation (1:1 MUX)

The 16-bit pixel port (A-B) is latched on the rising edge of GLCLK and is multiplexed 1:1. One rising edge of GLCLK should occur every pixel clock cycle.

Bit CR25 in Command Register 2 can be programmed to enable or disable true-color palette bypass. When the bypass mode is selected, the pixel data bypasses the palette as well as the pixel mask, and is transferred to the proper MSBs of the respective DACs. When the bypass mode is not selected, the pixel data indexes the proper locations in the palette, and the correct color information is passed to the respective DACs. Bit CR22 in Command Register 2 determines whether palette addressing is sparse or contiguous. For sparse palette addressing, each independent color component of pixel data is mapped to the most significant bits of the respective palette address; the least significant bits are set to zero. For contiguous palette addressing, each independent color component of the pixel data is mapped to the least significant bits of the respective palette address; the most significant bits are set to zero. The color palette values indexed, for either sparse or

contiguous addressing, are transferred to the DACs. When 5:5:5 or 5:6:5 color format is selected, the display can contain 32K or 64K simultaneous colors. The DACs can be configured for 6 or 8 bits of resolution in this mode (see Tables 3-4).

If 5:5:5 color format is selected, the most significant bit may be used for color key operation.

## 24-Bits/Pixel Operation (1:1 MUX)

When 24 bits per pixel in 1:1 MUX mode is selected, there is one 24-bit pixel port, (A-C). The pixel bits are latched on the rising edge of GLCLK and multiplexed 1:1. One rising edge of GLCLK should occur every pixel clock cycle. The RGB color format in this mode is 8:8:8.

Bit CR25 in Command Register 2 can be programmed to enable or disable true-color palette bypass. When the bypass mode is selected, the pixel data bypasses the palette as well as the pixel mask, and is transferred to the proper MSBs of the respective DACs. When the bypass mode is not selected, the pixel data indexes the proper locations in the palette, and the independent RGB color values are passed to the respective DACs (see Table 6a-6c). When 8:8:8 color format is selected, the display can contain 16.8 million simultaneous colors. The DACs should be configured for 8 bits of resolution in this mode (CR25 = 1, CR01 = 1). CR41 and CR40 can be used to alter the pixel read order to BRG or BGR.

## Pixel Read Mask Register

The pixel data can be masked before being transferred to the color palette with the 8-bit pixel mask register. The pixel data is bit-wise logically ANDed with the contents of the pixel read mask register. The result is used to address the color palette RAM. The addressed location provides 24 bits of color information to the three D/A converters. Pixel masking is enabled for all modes of operation except when the true-color bypass is enabled. The pixel mask register is initialized to logical ones at reset (see Table 13, Register Values on Reset in the Internal Register section).

## Circuit Description (continued)

Bit	MSB															LSB
Format	X	R	R	R	R	R	G	G	G	G	G	B	B	B	B	B
Port 1	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A
Port 2	P7D	P6D	P5D	P4D	P3D	P2D	P1D	P0D	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C

Note: X bit may be used for color key before the palette.

Table 3. 5:5:5 RGB Graphics Color Format for Both 2:1 and 1:1 Multiplexing Modes.

Bit	MSB															LSB
Format	R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B
Port 1	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A
Port 2	P7D	P6D	P5D	P4D	P3D	P2D	P1D	P0D	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C

Table 4. 5:6:5 RGB Graphics Color Format for Both 2:1 and 1:1 Multiplexing Modes.

	MSB							LSB	X=Map to Zero
Pixel Mask Register	7	6	5	4	3	2	1	0	Register Bits
4 Bits/Pixel	x	x	x	x	3	2	1	0	Palette Index
8 Bits/Pixel	7	6	5	4	3	2	1	0	Palette Index
16 Bits/Pixel	7	6	5	4	3	x	x	x	Red Palette Index
5:5:5 Format	7	6	5	4	3	x	x	x	Green Palette Index
SPARSE	7	6	5	4	3	x	x	x	Blue Palette Index
16 Bits/Pixel	x	x	x	4	3	2	1	0	Red Palette Index
5:5:5 Format	x	x	x	4	3	2	1	0	Green Palette Index
CONTIGUOUS	x	x	x	4	3	2	1	0	Blue Palette Index
16 Bits/Pixel	7	6	5	4	3	x	x	x	Red Palette Index
5:6:5 Format	7	6	5	4	3	2	x	x	Green Palette Index
SPARSE	7	6	5	4	3	x	x	x	Blue Palette Index
16 Bits/Pixel	x	x	x	4	3	2	1	0	Red Palette Index
5:6:5 Format	x	x	5	4	3	2	1	0	Green Palette Index
CONTIGUOUS	x	x	x	4	3	2	1	0	Blue Palette Index
24 Bit/Pixel	7	6	5	4	3	2	1	0	Red Palette Index
8:8:8 Format	7	6	5	4	3	2	1	0	Green Palette Index
	7	6	5	4	3	2	1	0	Blue Palette Index

Note: x means final DAC bit will be 0

Table 5. Graphics Pixel Index Masking.

Circuit Description (continued)

BH	MSB																							LSB
Format	R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B
Port 1	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A

Table 6a. 24-bits/Pixel Graphics RGB Color Format (CR41,40= 00) for 1:1 MUX Modes.

BH	MSB																							LSB
Format	B	B	B	B	B	B	B	B	R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G
Port 1	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A

Table 6b. 24-bits/Pixel Graphics BRG Color Format (CR41,40 = 01).

BH	MSB																							LSB
Format	B	B	B	B	B	B	B	B	G	G	G	G	G	G	G	G	R	R	R	R	R	R	R	R
Port 1	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A

Table 6c. 24-bits/Pixel Graphics BGR Color Format (CR41,40 = 10).

**Circuit Description (continued)****Modes of Operation—Video**

The pixel ordering and YCrCb-to-RGB conversions are shown in Figures 3 through 5, and the video pixel port configuration is shown in Table 11 in the Internal Registers section. The following describes video operation modes.

**YCrCb 1:0.5:0.5 Operation (4 Bytes/8 Pixels)**

The 32 input bits are configured for YCrCb 1:0.5:0.5. There are four independent 8-bit pixel ports, (E–H). Each group of four bytes results in eight output pixels. The pixel bits are latched on the rising edge of VLCLK.

Bit	MSB																LSB
Format	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Y	Y	Y	Y	Y	Y	Y	Y	
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	
Format	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Y	Y	Y	Y	Y	Y	Y	Y	
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	

**YCrCb 1:0.5:0.5 Video Color Format (4 Bytes / 8 Pixels).****YCrCb 1:0.5:0.5 Operation (2 Bytes/4 Pixels)**

The 16 input bits are configured for YCrCb 1:0.5:0.5. There are two independent 8-bit pixel ports, (G–H). Each group of two bytes results in four output pixels. The pixel bits are latched on the rising edge of VLCLK.

Bit	MSB																LSB
Format	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Y	Y	Y	Y	Y	Y	Y	Y	
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	

**YCrCb 1:0.5:0.5 Video Color Format (2 Bytes / 4 Pixels).****YCrCb 1:0.5:0.5 Operation (1 Byte/2 Pixels)**

The 8 input bits are configured for YCrCb 1:0.5:0.5. There is one 8-bit pixel port, (H). Each byte loaded results in two output pixels. The pixel bits are latched on the rising edge of VLCLK.

Bit	MSB																LSB
Format	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	X	X	X	X	X	X	X	X	
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	

**YCrCb 1:0.5:0.5 Video Color Format (1 Byte / 2 Pixels).**

**Circuit Description (continued)****YCrCb 2:1:1 Operation (4 Bytes/4 Pixels)**

The 32 input bits are configured for YCrCb 2:1:1. There are four independent 8-bit pixel ports, (E-H). The pixel bits are latched on the rising edge of VLCLK.

Bit	MSB																LSB
Format	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Y	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	
Format	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Y	Y	Y	Y	Y	Y	Y	Y	Y
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	

**YCrCb 2:1:1 Video Color Format (4 Bytes / 4 Pixels).****YCrCb 2:1:1 Operation (2 Bytes/2 Pixels)**

The 16 input bits are configured for YCrCb 2:1:1. There are two independent 8-bit pixel ports, (G-H). The pixel bits are latched on the rising edge of VLCLK.

Bit	MSB																LSB
Format	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Y	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	

**YCrCb 2:1:1 Video Color Format (2 Bytes / 2 Pixels).****YCrCb 2:1:1 Operation (1 Byte/1 Pixel)**

The 8 input bits are configured for YCrCb 2:1:1. There is one 8-bit pixel port, (H). The pixel bits are latched on the rising edge of VLCLK.

Bit	MSB																LSB
Format	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	Cb/Y/ Cr/Y	X	X	X	X	X	X	X	X	X
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	

**YCrCb 2:1:1 Video Color Format (1 Byte / 1 Pixel).**

## Circuit Description (continued)

## YCrCb 4:2:2 Operation (4 Bytes/2 Pixels)

The 32 input bits are configured for 4:2:2. There are two independent 16-bit pixel ports, (F-E) and (H-G). The bits are latched on the rising edge of VLCLK.

Bk	MSB															LSB						
Format	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G						
Format	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E						

YCrCb 4:2:2 Video Color Format (4 Bytes / 2 Pixels).

## YCrCb 4:2:2 Operation (2 Bytes/1 Pixel)

The 16 input bits are configured for YCrCb 4:2:2. There is one 16-bit pixel port, (G-H). The input bits are latched on the rising edge of VLCLK.

Bk	MSB															LSB						
Format	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G						
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E						

YCrCb 4:2:2 Video Color Format (2 Bytes / 1 Pixel).

## 16-Bits/Pixel 5:5:5 Operation (2:1 MUX)

The 32 input bits are configured for 16 bits/pixel. There are two independent 16-bit pixel ports, (E-F) and (G-H). The bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:5:5. The most significant bit is not used.

## 16-Bits/Pixel 5:5:5 Operation (1:1 MUX)

The 16 input bits are configured for 16 bits/pixel. There is one 16-bit pixel port, (G-H). The input bits are latched on the rising edge of VLCLK. The RGB Color format in this mode is 5:5:5. The most significant bit is not used.

Bk	MSB															LSB						
Format	X	R	R	R	R	R	G	G	G	G	G	B	B	B	B	B	B	B	B	B	B	B
Port 1	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H						
Port 2	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F						

5:5:5 RGB Video Color Format for Both 2:1 and 1:1 Multiplexing Modes.

**Circuit Description (continued)****16-Bits/Pixel 5:6:5 Operation (2:1 MUX)**

The 32 input bits are configured for 16 bits/pixel. There are two independent 16-bit pixel ports, (E-F) and (G-H). The bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:6:5.

**6-Bits/Pixel 5:6:5 Operation (1:1 MUX)**

The 16 input bits are configured for 16 bits/pixel. There is one 16-bit pixel port, (G-H). The input bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:6:5.

Bit	MSB															LSB
Format	R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B
Port 1	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H
Port 2	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F

**5:6:5 BGR Video Color Format for Both 2:1 and 1:1 Multiplexing Modes.**

**24-Bits/Pixel Operation (1:1 MUX)**

The 24 input bits are configured for 24 bits/pixel. There is one 24-bit pixel port, (F-H). The bits are latched on the rising edge of VLCLK. The RGB or BGR color format in this mode is 8:8:8. The color format is controlled by bit CR44 in Command Register 4.

Bit	MSB															LSB
Format	R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G
Port 1	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	B	B	B	B	B	B	B	B								
Port 2	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F

**24-Bit RGB Video Color Format (CR44=0) for 1:1 Multiplexing Modes.**

Bit	MSB															LSB
Format	B	B	B	B	B	B	B	B	G	G	G	G	G	G	G	G
Port 1	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	R	R	R	R	R	R	R	R								
Port 2	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H								

**24-Bit BGR Video Color Format (CR44=1) for 1:1 Multiplexing Modes.**

## Circuit Description (continued)

CCIR601 1:0.5:0.5

CCIR656 Component Ordering

Color Space: YCrCb

Subsampling: 1:0.5:0.5

Address (8N+)	0	1	2	3	4	5	6	7
Value	Cb0	Y0	Cr0	Y4	Cb8	Y8	Cr8	Y12

Pixel	0	1	2	3	4	5	6	7
Y	Y0	$\frac{3Y0 + Y4}{4}$	$\frac{Y0 + Y4}{2}$	$\frac{Y0 + 3Y4}{4}$	Y4	$\frac{3Y4 + Y8}{4}$	$\frac{Y4 + Y8}{2}$	$\frac{Y4 + 3Y8}{4}$
Cr	Cr0	Cr0	$\frac{3Cr0 + Cr8}{4}$	$\frac{3Cr0 + Cr8}{4}$	$\frac{Cr0 + Cr8}{2}$	$\frac{Cr0 + Cr8}{2}$	$\frac{Cr0 + 3Cr8}{4}$	$\frac{Cr0 + 3Cr8}{4}$
Cb	Cb0	Cb0	$\frac{3Cb0 + Cb8}{4}$	$\frac{3Cb0 + Cb8}{4}$	$\frac{Cb0 + Cb8}{2}$	$\frac{Cb0 + Cb8}{2}$	$\frac{Cb0 + 3Cb8}{4}$	$\frac{Cb0 + 3Cb8}{4}$

Figure 3. CCIR601 1:0.5:0.5 Video Format.

CCIR601 2:1:1

CCIR656 Component Ordering

Color Space: YCrCb

Address (8N+)	0	1	2	3	4	5	6	7
Value	Cb0	Y0	Cr0	Y2	Cb4	Y4	Cr4	Y6

Pixel	0	1	2	3	4	5	6	7
Y	Y0	$\frac{Y0 + Y2}{2}$	Y2	$\frac{Y2 + Y4}{2}$	Y4	$\frac{Y4 + Y6}{2}$	Y6	$\frac{Y6 + Y8}{2}$
Cr	Cr0	$\frac{3Cr0 + Cr4}{4}$	$\frac{Cr0 + Cr4}{2}$	$\frac{Cr0 + 3Cr4}{4}$	Cr4	$\frac{3Cr4 + Cr8}{4}$	$\frac{Cr4 + Cr8}{2}$	$\frac{Cr4 + 3Cr8}{4}$
Cb	Cb0	$\frac{3Cb0 + Cb4}{4}$	$\frac{Cb0 + Cb4}{2}$	$\frac{Cb0 + 3Cb4}{4}$	Cb4	$\frac{3Cb4 + Cb8}{4}$	$\frac{Cb4 + Cb8}{2}$	$\frac{Cb4 + 3Cb8}{4}$

Figure 4. CCIR601 2:1:1 Video Format.



## Circuit Description (continued)

CCIR601 4:2:2

CCIR656 Component Ordering

Color Space: YCrCb

Address (8N+)	0	1	2	3	4	5	6	7
Value	Cb0	Y0	Cr0	Y1	Cb2	Y2	Cr2	Y3

Ptzel	0	1	2	3
Y	Y0	Y1	Y2	Y3
Cr	Cr0	$\frac{Cr0 + Cr2}{2}$	Cr2	$\frac{Cr2 + Cr4}{2}$
Cb	Cb0	$\frac{Cb0 + Cb2}{2}$	Cb2	$\frac{Cb2 + Cb4}{2}$

Figure 5. CCIR601 4:2:2 Video Format.

**DAC Values in 16-Bits/Pixel Video Modes**

In order to achieve 8-bit full-scale DAC output in the 5:5:5 16-bits/pixel video modes, each 5-bit value will be used as the five most significant bits of the 8-bit DAC value and the three most significant bits of the 5-bit pixel value will be duplicated in the low order 3 bits

before the pixel value is passed to the DACs. Similarly, in 5:6:5 modes, when processing the 6-bit green component, the 6-bit value will be used as the 6 most significant bits of the 8-bit DAC value and the two most significant bits of the 6-bit pixel value will be duplicated in the low order 2 bits before the pixel value is passed to the DACs.

## Circuit Description (continued)

## Cursor Operation

The Bt885 has an on-chip, three-color,  $64 \times 64 \times 2$  pixel user-definable cursor. This cursor works with both interlaced and noninterlaced systems. The cursor always has display priority over both video and graphics pixels.

The pattern for the cursor is provided by the cursor RAM, which may be accessed by the MPU at any time. The cursor is positioned through the cursor position register (Xp,Yp) (see Figure 6). A (0,0) written to the cursor position registers will place the cursor completely offscreen. A (1,1) written to the cursor position registers will place the lower right pixel of the cursor on the upper left corner of the screen. Only one cursor pattern per frame is displayed at the location specified for both interlaced and noninterlaced display formats, regardless of the number of updates to (Xp,Yp). The cursor's vertical or horizontal location is not affected during any frame displayed.

There are no restrictions on updating (Xp, Yp) other than both cursor position registers must be written when the cursor location is updated. Internal x and y position registers are loaded after the upper byte of Yp has been

written to ensure one cursor pattern per frame at the correct location. The cursor pattern is displayed at the last cursor location written. Cursor positioning is relative to ENABLE. The cursor position is not dependent upon BLANK\* (see Figure 6). The cursor Xp position is relative to the first rising edge of GLCLK when ENABLE is sampled at logical one. The cursor Yp position is relative to the first rising edge of GLCLK when ENABLE is sampled at logical one after the ENABLE vertical blanking interval has been determined (see Figure 6). If an ENABLE transition from logical zero to logical one (as determined by GLCLK) does not occur within 2048 internal pixel clocks, ENABLE is in vertical blanking.

For proper cursor operation, selection of interlaced or non-interlaced cursor display must be set using bit CR23 in Command Register 2.

Figure 7 is a visual explanation of planar pixel format and cursor RAM array pixel mapping.

While the cursor may be disabled by setting bits CR20-21 of Command Register 2 to zero, this practice is not recommended. The recommended method for disabling the cursor is to move it entirely offscreen by setting the cursor X and Y location registers to (0,0).

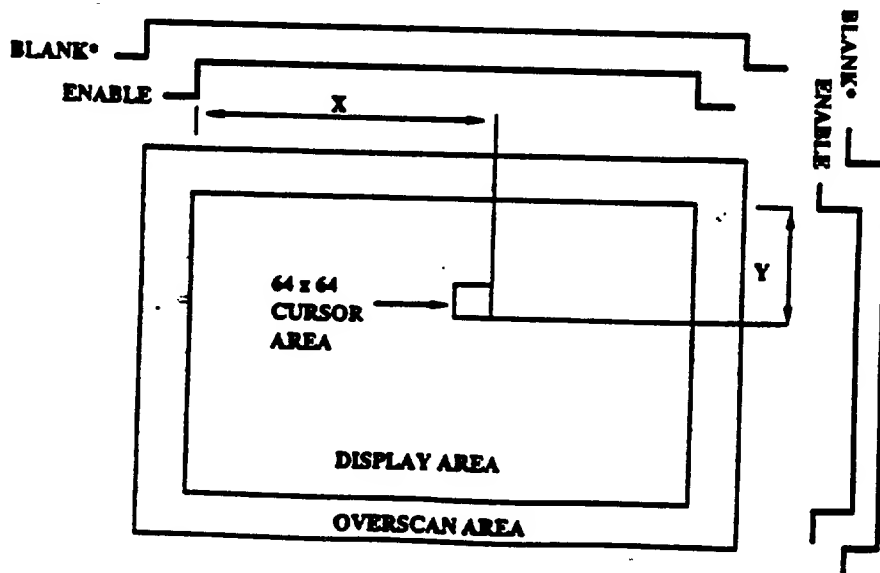


Figure 6. Cursor Positioning.

## Circuit Description (continued)

## Cursor Color Support

The cursor has three modes for color selection. Bits CR21 and CR20 in Command Register 2 determine which cursor mode is to be used. Mode 1 is a three-color cursor. Mode 2 is a Microsoft Windows™ cursor, and Mode 3 is an X-Windows cursor (see Table 7).

## Highlight Logic

The highlight logic is enabled in cursor mode 2 when plane 1 and plane 0 data are logical ones (see Table 7). When the highlight logic is enabled, it ensures that the graphics pixel highlighted has a unique color. This is because the highlight logic bit-wise complements the 24 (18)-bit graphics palette or bypass data supplied to the DACs.

## Video Generation

The C/HSYNC\* and BLANK\* inputs are latched on the rising edge of GLCLK to maintain synchronization with the pixel data.

Pipelined C/HSYNC\* and VSYNC\* are output on the C/HSYNC\* OUT and VSYNC\* OUT pins.

The CR05 command bit specifies whether a 0 or 7.5 IRE blanking pedestal is to be used. Command bits CR04, CR03, and CR02 specify whether the RGB outputs contain sync information.

Figures 8 and 9, and Tables 8 and 9 detail how the C/HSYNC\* and BLANK\* inputs modify the output levels.

## SENSE\* Output

SENSE\* is a logical zero if one or more of the IOR, IOG, or IOB outputs have exceeded the internal voltage reference level of the SENSE\* comparator circuit. This output determines the presence of a CRT monitor and, with diagnostic code, the difference between a loaded or an unloaded RGB line can be discerned. The reference is generated by a voltage divider from the external 1.235 V voltage reference on the VREF pin. For the proper operation of the SENSE circuit, the following levels should be applied to the comparator with the IOR, IOG, and IOB outputs:

DAC Low Voltage  $\leq 260$  mV (see note below)

DAC High Voltage  $\geq 410$  mV (see note below)

There is an additional  $\pm 10\%$  tolerance on the above levels when the internal voltage reference is used.

C/HSYNC\* should be a logical zero and BLANK\* should be a logical one for SENSE\* to be stable. The SENSE\* output can drive only one CMOS load.

*Note:* SENSE values are subject to change upon completion of characterization.

This is Pseudo Code for Bt885 to check for Monitor connection.  
Problem: Verify if an RGB or Single Input Monitor is connected to the RAMDAC Analog Outputs.

Program Monitor?	("Verify Monitor Connection")
Reset Bt885	("Toggle RESET* of Bt885")
Set C/HSYNC* = low	("Disable SYNC Current")
Set BLANK = high	("Enable RamDAC Outputs")
Set Pixel Mask = \$00	("Disable external Pixel Input")
Set RGB LUT Loc.0 = \$18	("10.3mV X 24 = 247mV")
Read Status Register	("Check on state of SENSE**")
IF SR1-3 = 1	("Check for RGB Monitor connection")
RGB Monitor	
ELSE	
Single Monitor?	
Address LUT Loc. \$00	("Set address register to program Lut Loc. 0")
Set Red DACOutput = \$00	("Set Red DAC to output 0mV")
Set Gm DACOutput = \$18	("Set Green DAC to output 247mV")
Set Blu DACOutput = \$00	("Set Blue DAC to output 0mV")
Read Status Register	("Check for Single Input Monitor on Gm.")
IF SR1-3 = 1	
Single Input Monitor	
ELSE	
NO Monitor Sensed	
End	

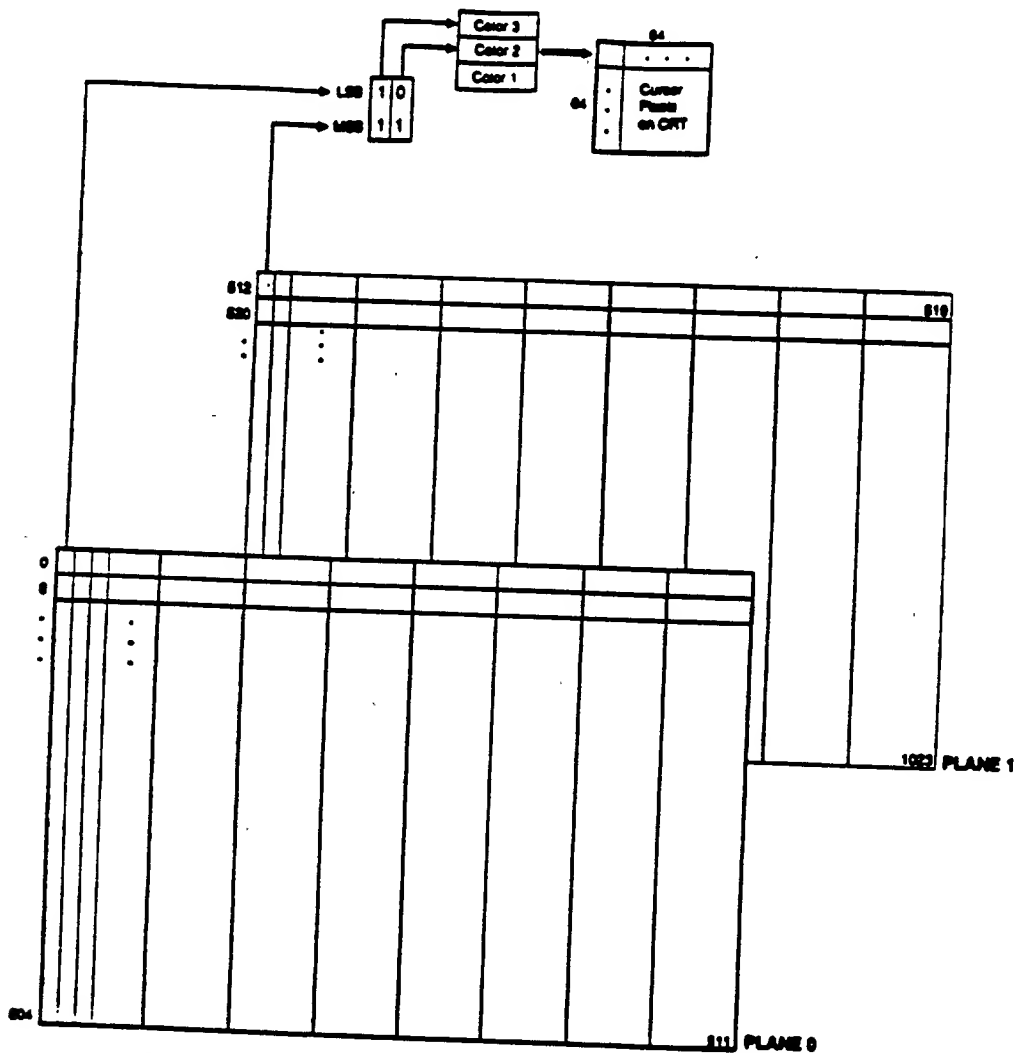
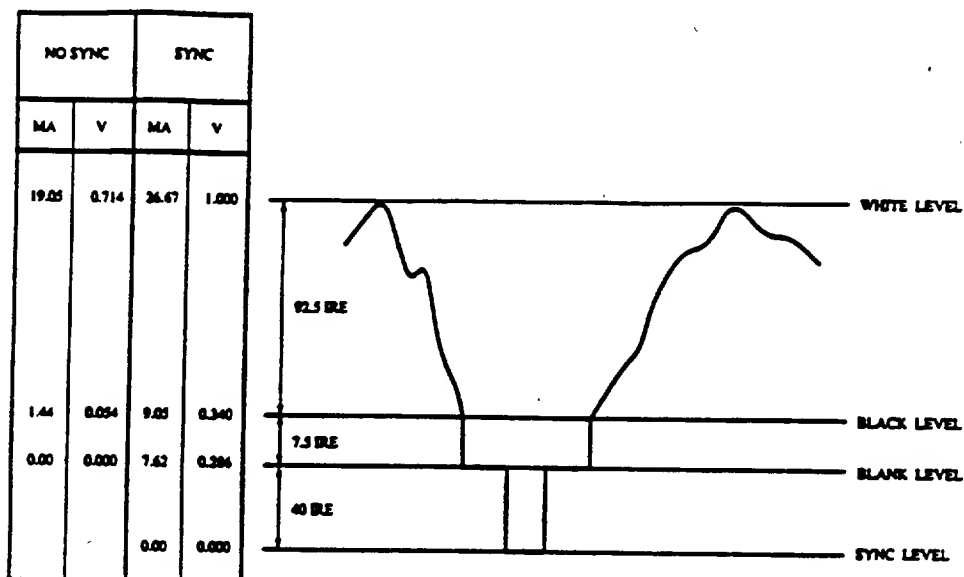


Figure 7. Planar Pixel Format and Cursor RAM Array Pixel Mapping.

Plane 1	Plane 0	MODE 1	MODE 2	MODE 3
0	0	Cursor not displayed	Cursor Color 1	Palette Data
0	1	Cursor Color 1	Cursor Color 2	Palette Data
1	0	Cursor Color 2	Palette Data	Cursor Color 1
1	1	Cursor Color 3	Highlight	Cursor Color 2

Table 7. Cursor Color Modes.

Circuit Description. (continued)



Note: 75  $\Omega$  doubly-terminated load, VREF = 1.235 V, and RSET = 147  $\Omega$ . RS-343A levels and tolerances assumed on all levels.

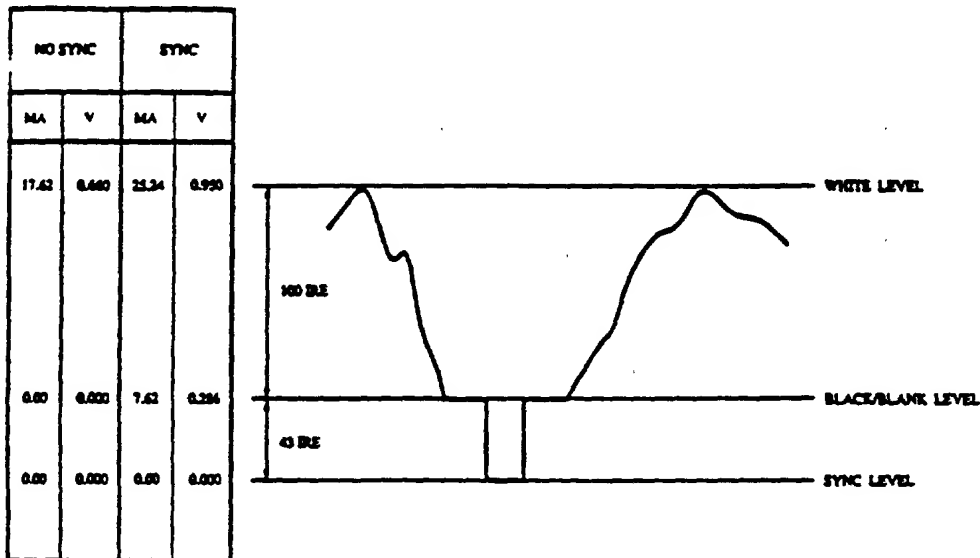
Figure 8. Composite Video Output Waveforms (SETUP = 7.5 IRE).

Description	Sync Disabled	Sync Enabled	C/HSYNC*	BLANK*	DAC Input Data
	Iout (mA)	Iout (mA)			
WHITE	19.05	26.67	1	1	0xFF
DATA	data + 1.44	data + 9.05	1	1	data
DATA-SYNC	data + 1.44	data + 1.44	0	1	data
BLACK	1.44	9.05	1	1	0x00
BLACK-SYNC	1.44	1.44	0	1	0x00
BLANK	0	7.62	1	0	xx
C/HSYNC*	0	0	0	0	xx

Note: 75  $\Omega$  doubly-terminated load, VREF = 1.235 V, and RSET = 147  $\Omega$ .

Table 8. Video Output Truth Table (SETUP = 7.5 IRE).

## Circuit Description (continued)



Note: 75  $\Omega$  doubly-terminated load, VREF = 1.235 V, and RSET = 147  $\Omega$ . RS-343A levels and tolerances assumed on all levels.

Figure 9. Composite Video Output Waveforms (SETUP = 0 IRE).

Description	Sync Disabled	Sync Enabled	CHSYNC*	BLANK*	DAC Input Data
	Iout (mA)	Iout (mA)			
WHITE	17.62	25.24	1	1	0xFF
DATA	data	data + 7.62	1	1	data
DATA-SYNC	data	data	0	1	data
BLACK	0	7.62	1	1	0x00
BLACK-SYNC	0	0	0	1	0x00
BLANK	0	7.62	1	0	xx
CHSYNC	0	0	0	0	xx

Note: 75  $\Omega$  doubly-terminated load, VREF = 1.235 V, and RSET = 147  $\Omega$ .

Table 8. Video Output Truth Table (SETUP = 0 IRE).

## Internal Registers

## Command Register 0

This register may be written to or read by the MPU at any time and is not initialized at power-up. CR00 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET™ pin.

CR0 7	Reserved	This bit must be written with a 0 to ensure proper operation.
CR0 6	Clock Disable ANDed with CR00 (0) Normal Operation (1) Disable Internal Clocking	When this bit and CR00 are a logical one, the internal clock and output clocks are disabled to further conserve power when in power-down mode. The RAM still retains the data, and MPU reads and writes can occur with no loss of data. When this bit is a logical zero, internal clocking is enabled and output clocks will be generated.
CR0 5	Pedestal IRE (0) Disable (1) Enable 7.5 IRE	This bit determines the video blanking pedestal. A logical zero always sets a 0 IRE blanking pedestal and a logical one sets 7.5 IRE.
CR0 4	Blue Sync Enable	These bits specify whether the respective IOB, IOG, or IOR outputs are to contain sync information.
CR0 3	Green Sync Enable	
CR0 2	Red Sync Enable (0) Disable Sync (1) Enable Sync	
CR0 1	DAC 6/8-Bit Resolution (0) 6-bit Operation (1) 8-bit Operation	This bit specifies whether the MPU is reading and writing 8 bits (logical one) or 6 bits (logical zero) of color information each cycle.
CR0 0	Power-Down Enable (0) Normal Operation (1) Power-Down Operation	While this bit is a logical zero, the device operates normally. If this bit is a logical one, the DACs and power to the RAM and VideoCache™ FIFO are turned off. The RAM still retains the data, and CPU reads and writes can occur with no loss of data.  The DACs will be turned off during sleep mode only if a voltage reference (internal or external) is used.

**Internal Registers (continued)****Command Register 1**

This register may be written to or read by the MPU at any time. CR10 corresponds to data bus bit D0, the least significant data bit (see Table 10). All command register bits are set to logical zero upon asserting a low signal on the RESET\* pin.

CR17	CR16	CR15	CR14	Pixel Latching Sequence	Bytes Per VLCLK	Pixels Per VLCLK	Operating Modes
0	0	0	0	N/A	N/A	N/A	All Video Modes Disabled
0	0	0	1	P7:0(H)	1	2	CCIR601 YCrCb 1:0.5:0.5
0	0	1	0	P7:0(H) P7:0(G)	2	4	CCIR601 YCrCb 1:0.5:0.5
0	0	1	1	P7:0(H) P7:0(G) P7:0(F) P7:0(E)	4	8	CCIR601 YCrCb 1:0.5:0.5
0	1	0	0	P7:0(H)	1	1	CCIR601 YCrCb 2:1:1
0	1	0	1	P7:0(H) P7:0(G)	2	2	CCIR601 YCrCb 2:1:1
0	1	1	0	P7:0(H) P7:0(G) P7:0(F) P7:0(E)	4	4	CCIR601 YCrCb 2:1:1
0	1	1	1	P7:0(H-G)	2	1	CCIR601 YCrCb 4:2:2
1	0	0	0	P7:0(H-G) P7:0(F-E)	4	2	CCIR601 YCrCb 4:2:2
1	0	0	1	P7:0(H-G)	2	1	15-bits per pixel, 5:5:5
1	0	1	0	P7:0(H-G) P7:0(F-E)	4	2	15-bits per pixel, 5:5:5
1	0	1	1	P7:0(H-G)	2	1	16-bits per pixel, 5:6:5
1	1	0	0	P7:0(H-G) P7:0(F-E)	4	2	16-bits per pixel, 5:6:5
1	1	0	1	P7:0(H-F)	3	1	24-bits per pixel
1110-1111							Reserved

**Table 10. Modes of Operation (Video Pixel Port Configuration).**



## Internal Registers (continued)

All command register bits are set to logical zero upon asserting a low signal on the RESET® pin.

CR13	CR12	CR11	CR10	Pixel Latching Sequence	MUX Rate	Operating Modes
0	0	0	0	P7:0(A)	1:1	VGA 8-bits per pixel
0	0	0	1	P7:0(A) P7:0(B)	2:1	8-bits per pixel
0	0	1	0	P7:0(A) P7:0(B) P7:0(C) P7:0(D)	4:1	8-bits per pixel
0	0	1	1	P7:4(A) P3:0(A) P7:4(B) P3:0(B) P7:4(C) P3:0(C) P7:4(D) P3:0(D)	8:1	4-bits per pixel
0	1	0	0	P7:0(B-A)	1:1	15-bits per pixel, 5:5:5
0	1	0	1	P7:0(B-A) P7:0(D-C)	2:1	15-bits per pixel, 5:5:5
0	1	1	0	P7:0(B-A)	1:1	16-bits per pixel, 5:6:5
0	1	1	1	P7:0(B-A) P7:0(D-C)	2:1	16-bits per pixel, 5:6:5
1	0	0	0	P7:0(C-A)	1:1	24-bits per pixel
1001-1111						Reserved

Table 11. Modes of Operation (Graphic Pixel Port Configuration).

**Internal Registers (continued)****Command Register 2**

This register may be written to or read by the MPU at any time. CR20 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET\* pin.

CR2 7	Reserved Logical 0	This bit must be written with a 0 to ensure correct operation.
CR2 6	Reserved Logical 0	This bit must be written with a 0 to ensure correct operation.
CR2 5	True-Color bypass Enable (0) Pixel Addresses Palette (1) Pixel Bypasses Palette	When this bit is a logical zero, the pixel palette is addressed by the pixel data. When this bit is a logical one, the RGB pixel data bypasses the color palette and drives the DACs directly. True-color bypassing is only available for pixel sizes of 16 and 24 bits.
CR2 4	Oscillator Select (0) OSC Selected (1) OSC* Selected	When this bit is a logical zero, OSC is selected as the TTL pixel clock input. When this bit is a logical one, OSC* is selected as the TTL pixel clock input.
CR2 3	Display Mode Select (0) Noninterlaced (1) Interlaced	When this bit is a logical zero, the display format is noninterlaced. When the bit is a logical one, the display format is interlaced. The mode must be set properly to ensure proper operation of the internal cursor.
CR2 2	16-Bit/Pixel Palette Index Select (0) Sparse Indexing (1) Contiguous Indexing	When this bit is a logical zero, palette addressing is sparse. The RGB color component pixel data is mapped to the most significant bits of the RGB palette address. The least significant of the palette address bits are set to (0). When this bit is a logical one, palette addressing is contiguous. The RGB color component pixel data is mapped to the least significant bits of the palette address. The most significant bits of the address are set to (0).
CR2 1.0	Cursor Mode Select (00) Cursor Disabled (01) 3-color cursor (10) 2-color/Microsoft Windows™ cursor (11) 2-color/X-Windows cursor	These bits determine the functionality of the onboard 64 x 64 x 2 hardware cursor.

**Internal Registers (continued)****Accessing the Extended Registers**

An extended register set is used to accommodate all features of the Bt885. Since there are only four register select lines (and all 16 combinations have already been used), the extended registers must be accessed indirectly.

For example, Command Register 3 is accessed with the following sequence of operations:

1. Set RS3-RS0 = 0000, Address Register.
2. Write Address Register to 0x02
3. Set RS3-RS0 = 1010 (Extended Address Register).
4. Read or Write Command Register 3.

Table 12 shows the indirect addressing mapping for each extended register.

Address Register Value	Extended Register Name
0x00	Status Register 1 (read only)
0x01	Status Register 2 (read/write)
0x02	Command Register 3
0x03	Command Register 4
0x04 - 0x05	Video Window XSTART—Low & High
0x06 - 0x07	Video Window YSTART—Low & High
0x08 - 0x09	Video Window XWIDTH—Low & High
0x0A - 0x0B	Video Window YHEIGHT—Low & High
0x0C - 0x0D	Reserved
0x0E - 0x0F	Reserved
0x10 - 0x11	XSCALEINT—Low & High
0x12 - 0x13	XSCALEINC—Low & High
0x14 - 0x15	Reserved
0x16 - 0x17	Reserved
0x18 - 0x19	Serial Clock Enable Start (Horizontal)—Low & High
0x1A - 0x1B	Serial Clock Enable Duration (Horizontal)—Low & High
0x1C - 0x1D	Reserved
0x1E - 0x1F	Reserved
0x20	DIVCLK1 Rate
0x21	DIVCLK2 Rate
0x22	Reserved
0x23 - 0x25	Color Mask (Ordering = RGB)
0x26	Reserved
0x27 - 0x29	Color Key (Ordering = RGB)
0x2A - 0x2D	Reserved
0x2E	VideoCache™ FIFO Size
0x2F - 0xFF	Reserved

**Table 12. Extended Registers Address Map (RS3-RS0 = 1010).**

## Internal Registers (continued)

## Command Register 3

This register may be written to or read by the MPU at any time. CR30 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET<sup>®</sup> pin.

CR3 7	MODE0 Input/Output Select (0) MODE0 Input (1) MODE0 Output	This bit determines if the MODE0 pin is configured as an input or an output.
CR3 6	Enable Internal Load Clock (0) Use GLCLK (1) Use Internal DIVCLK2	In applications where an external load clock is not provided, setting CR36 = 1 allows the internal DIVCLK2, determined by the DIVCLK2 Register values, to internally sample the graphics input pixels, blanking, horizontal, and vertical sync inputs. Setting CR36 = 0 causes Bt885 to sample these inputs on the basis of GLCLK pin.
CR3 5	DIVCLK2 Select (0) DIVCLK2 Enabled (1) DIVCLK2 Disabled	A logical zero must be written to this bit to enable the graphics divide-down clock, DIVCLK2, to be output. A logical one written to this bit three-states the DIVCLK2 output.
CR3 4	ECL Clock Select (0) TTL Level Clock Selected (1) Differential ECL Level Clock Selected	A logical one written to this bit enables the differential ECL clock input buffer using OSC and OSC <sup>®</sup> as inputs. A logical zero written to this bit disables the ECL clock buffer and allows OSC, GLCLK, or the 2x clock multiplier to directly drive the logic. If a logical one is written to this bit, then the clock multiplier and TTL clock selections are overridden. If CR34 = 1, then bit CR33 must be set to zero.
CR3 3	2x Clock Multiplier Select (0) 2x Clock Multiplier Disabled. (1) 2x Clock Multiplier Enabled.	This bit enables or disables the 2x clock multiplier. A logical one written to this bit enables the onboard 2x TTL clock multiplier for high-speed operations. A logical zero written to this bit will disable the clock multiplier and will allow the external clock source to directly drive the logic. If CR34 = 1, then this bit must be set to zero.
CR3 2	DIVCLK1 Select (0) DIVCLK1 Enabled (1) DIVCLK1 Disabled	A logical zero must be written to this bit to enable the video divide-down clock, DIVCLK1, to be output. A logical one written to this bit three-states the DIVCLK1 output. If DIVCLK1 Select is set to one, then the SEN output pin is three-stated as well.
CR3 1,0	MSBs for 10-bit Address Counter CR31 = A9 CR30 = A8	CR31 and CR30 are 2 MSBs of the 10-bit cursor address counter. To set this counter to access a particular location in the 64 x 64 x 2 cursor RAM array, these 2 bits must be written to Command Register 3 before the lower 8 bits are written to the address counter through the MPU port. As the 10-bit address counter autoincrements, the new values of this register can be read back through CR31 and CR30.

## Internal Registers (continued)

## Command Register 4

This register may be written to or read by the MPU at any time and is not initialized at power-up. CR40 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET® pin.

CR4	7	VideoCache™ FIFO Reset (0) Normal Operation (1) Reset VideoCache™ FIFO	A logical zero written to this bit, enables normal VideoCache™ FIFO operation. A logical one written to this bit resets the VideoCache™ FIFO after four video load clocks.
CR4	6	Color Key Override (0) Normal Color Key Operation (1) Video Window Override	A logical zero written to this bit, enables standard color key operation. A logical one written to this bit enables video based only on the video window.
CR4	5	Set MODE0 State (CR37 = 1) (0) MODE0 pin low (1) MODE0 pin high  Get MODE0 State (CR37 = 0) (0) MODE0 pin externally driven low (1) MODE0 pin externally driven high	When CR37 = 1, this bit controls the state of the MODE0 output. A logical one written to this bit sets the MODE0 pin to high. A logical zero written to this bit sets the MODE0 pin low.  When CR37 = 0, this bit indicates the state of the MODE0 input. A logical one read from this bit indicates that the MODE0 pin is driven high. A logical one read from this bit indicates that the MODE0 pin is driven high. A logical zero read from this bit indicates that the MODE0 pin is driven low.
CR4	4	24-bit Video Component Order (0) RGB (1) BGR	This bit controls the component latching order in 24-bit per-pixel video modes.
CR4	3	Color Key Mode Select (0) Before Palette (1) After Palette	This bit controls whether color key matching occurs on the pixel value before or after the palette. A logical zero written to this bit selects color key matching on the pixel value before the palette. A logical one written to this bit selects color key matching on the 24-bit RGB value after the palette.
CR4	2	VideoCache™ Unload Select (0) Unload Within Video Window (1) Unload From Start of Active Graphics Enable	This bit controls whether VideoCache™ FIFO data is unloaded only within the video window or at all times during active graphics enable.
CR4	1,0	24-bit Graphics Component Order (00) RGB (01) BRG (10) BGR (11) Reserved	This bit controls the component latching order in 24-bit-per-pixel graphic modes. If any other graphics mode is selected, these bits must be set to logical zero.

**Internal Registers (continued)****Pixel Read Mask Register**

The 8-bit pixel read mask register may be written to or read by the MPU at any time, and is initialized to 0xFF at power-up. D0 is the least significant bit. The contents of this register are bit-wise ANDed with the pixel data prior to addressing the color palette RAM.

**Status Registers 1-2**

These two 8-bit status registers are provided for device identification and to monitor certain device states. They may be read by the MPU at any time. MPU write cycles to status register 1 are ignored. D0 is the least significant bit corresponding to SR10 or SR20. These registers are not reset during power-up/reset.

SR1	7-6	Chip Identification	These bits are identification values; SR17 = 1 and SR16 = 0.
SR1	5-4	Chip Revision	These bits are revision values; SR15 = 1 and SR14 = 0.
SR1	3	Monitor Sense	This is the SENSE* bit. If it is a logical zero, one or more of the IOR, IOG, and IOB outputs have exceeded the internal voltage reference level (335 mV). This bit is used to determine the presence of a CRT monitor and, with diagnostic code, the difference between a loaded or unloaded RGB line can be discerned. The 360 mV reference has a $\pm 100$ mV tolerance when an external voltage reference equal to 1.235 V is used. A greater tolerance is expected when an internal reference equal to 1.2 V is used.
SR1	2	Read/Write access status. (0) Write Cycle (1) Read Cycle	This bit provides RD/WR status when Address Register 0x00, 0x03, 0x04, or 0x07 has been written. When Address Register 0x00 or 0x04 has been written, the device is in the write mode and this bit is a logical zero. When address register 0x03 or 0x07 has been written, the device is in the read mode and this bit is a logical one.
SR1	1-0	RGB Component Counter (00) Red Color Component (01) Green Color Component (10) Blue Color Component	When read, these bits reflect the color component address for the next RD/WR cycle when accessing the palette, cursor color registers, or overscan register.
SR2	7	VideoCache™ FIFO Underflow	Reading this bit as a one indicates that VideoCache™ FIFO underflow occurred. Reset by writing any value to Status Register 2.
SR2	6-0	Reserved	These bits will always be read as zero.

## Internal Registers (continued)

### Video Window XSTART

Video Window XSTART is a 12-bit register that stores the starting X position on the screen for a video window. A value of zero indicates that the video window begins in the first (leftmost) pixel of each horizontal scan line.

### Video Window YSTART

Video Window YSTART is a 12-bit register that stores the starting Y position on the screen for a video window. A value of zero indicates that the video window begins on the first active graphics scan line.

### Video Window XWIDTH

Video Window XWIDTH is a 12-bit register that stores the number of pixels per scan line within the video window. A value of zero indicates that no pixels are in the video window.

### Video Window YHEIGHT

Video Window YHEIGHT is a 12-bit register that stores the number of scan lines within the video window. A value of zero indicates that no scan lines are within the video window.

### XSCALEINIT (Low & High)

XSCALEINIT are 12-bit registers that store the initial term for the horizontal scaler.

### XSCALEINC (Low & High)

XSCALEINC are 12-bit registers that store the increment term for the horizontal scaler.

### Serial Clock Enable Start (Horizontal)

Serial clock enable start (horizontal and vertical) are 12-bit registers that store the number of scan lines and DIVCLK1 cycles before enabling the external clock gate, starting at the leading edge of HSYNC\* for the horizontal direction and the leading edge of the internally generated VSYNC\* for the vertical direction.

### Serial Clock Enable Duration (Horizontal)

Serial clock enable duration (horizontal and vertical) are 12-bit registers that store the number of serial shift clock cycles to be generated per scan line in units of DIVCLK1 cycles for the horizontal direction, and in units of scan lines for the vertical direction.

### DIVCLK1 and DIVCLK2 Rate

DIVCLK1 and DIVCLK2 rate are two 3-bit registers that control the divide rate of the free running DIVCLK1 and DIVCLK2 divide-down clocks, respectively. The divide-down ratios need not be the same as the input mux rate:

- (000) - 1:1
- (001) - 2:1
- (010) - 4:1
- (011) - 8:1
- (100-111) - Reserved

### VideoCache™ FIFO Size

This register indicates the length of the VideoCache™ FIFO buffer in 16-byte units. This is a read-only register.

## Internal Registers (continued)

## Cursor (x,y) Registers

These registers are used to specify the (x,y) coordinate of the 64 x 64 x 2 hardware cursor. The cursor (x) register is made up of the cursor (x) low register (CXLR) and the cursor (x) high register (CXHR); the cursor (y) register is made up of the cursor (y) low register (CYLR) and the cursor (y) high register (CYHR). The last value written by the MPU to these registers is the value returned on a read. These registers may be written to or read by the MPU at any time.

CXLR and CXHR are cascaded to form a 12-bit cursor (x) register. Similarly, CYLR and CYHR are cascaded to form a 12-bit cursor (y) register. Bits D4-D7 of CXHR and CYHR are ignored and should be written as zeros.

The cursor (x) value to be written is calculated as follows:

$$Xp = \text{desired display screen (x) position} + 64$$

where the (x) reference point for the display screen, x = 0, is the upper left corner of the screen. The Xp position

equation places the upper lefthand corner of the cursor RAM array to the desired screen location. This allows the cursor position to be defined in the same coordinate space as the screen.

Values from 0 to 4095 may be written into the cursor (x) register. If Xp is equal to zero, the cursor will be entirely offscreen.

The cursor (y) value to be written is calculated as follows:

$$Yp = \text{desired display screen (y) position} + 64$$

where the (y) reference point for the display screen, y = 0, is the upper left corner of the screen. The Yp position equation places the upper left corner of the cursor RAM array to the desired screen location. This allows the cursor position to be defined in the same coordinate space as the screen.

Values from 0 to 4095 may be written into the cursor (y) register. If Yp is equal to zero, the cursor will be entirely offscreen (see Cursor Operation in the Circuit Description section).

	Cursor (x) High (CXHR)				Cursor (x) Low (CXLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Address	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

	Cursor (y) High (CYHR)				Cursor (y) Low (CYLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Address	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0



## Internal Registers (continued)

Register Name	Reset Value
Command Register 0	0
Command Register 1	0
Command Register 2	0
Command Register 3	0
Command Register 4	0
Video Window XSTART—Low & High	Not Initialized
Video Window YSTART—Low & High	Not Initialized
Video Window XWIDTH—Low & High	Not Initialized
Video Window YHEIGHT—Low & High	Not Initialized
XSCALEINIT—Low & High	Not Initialized
XSCALEINC—Low & High	Not Initialized
Serial Clock Enable Start (Horizontal)—Low & High	Not Initialized
Serial Clock Enable Duration (Horizontal)—Low & High	Not Initialized
Serial Clock Enable Start (Vertical)—Low & High	Not Initialized
Serial Clock Enable Duration (Vertical)—Low & High	Not Initialized
DIVCLK1 Rate	0
DIVCLK2 Rate	0
Color Mask	0
Color Key	0
FIFO Size	0x32
Color Palette RAM	Not Initialized
Pixel Read Mask	0xFF
Cursor Colors	Not Initialized
Overscan Color	Not Initialized
Cursor X,Y	Not Initialized
Cursor RAM array	Not Initialized

Table 13. Register Values on Reset.

## Pin Descriptions

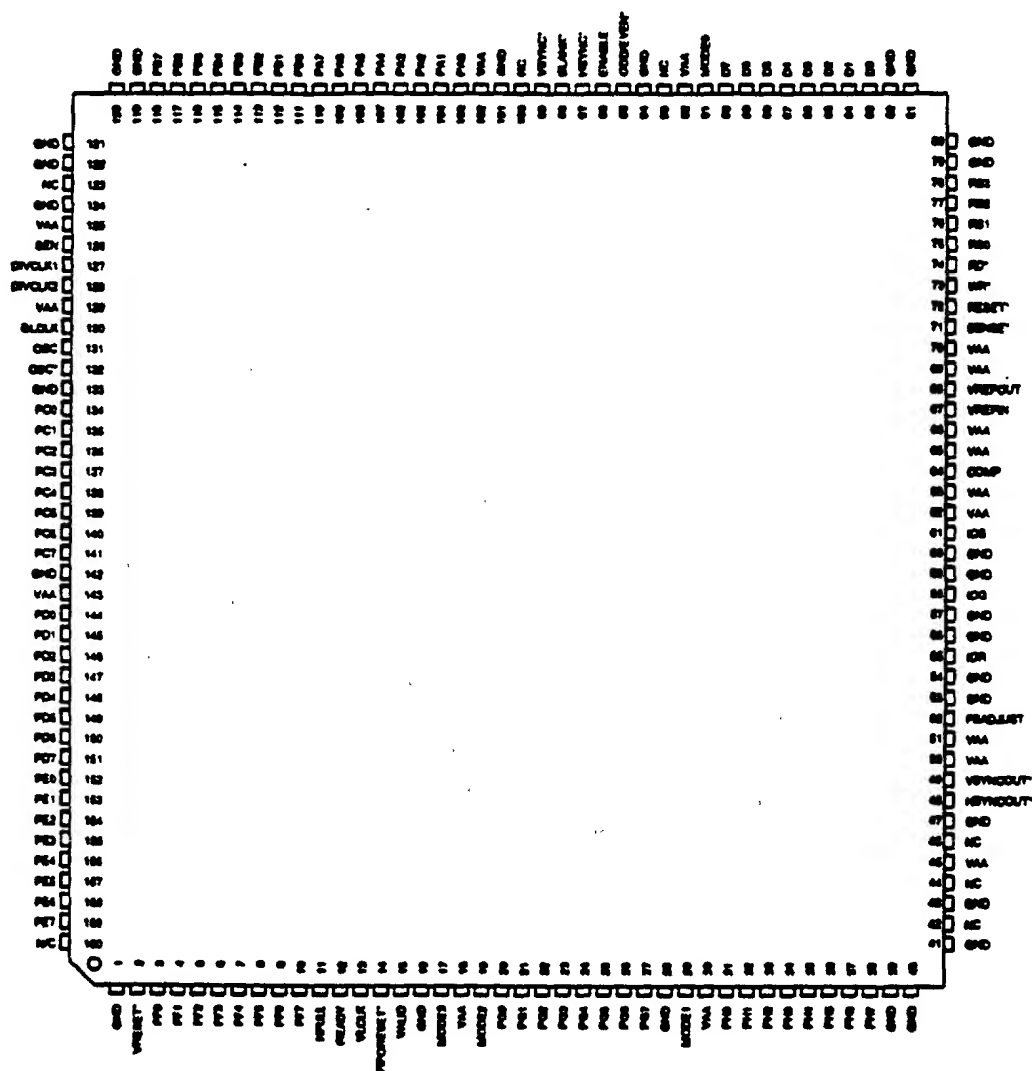
Pin Name	I/O	Pin#	Description												
RESET*	I	72	Reset input (TTL compatible). When this signal is low, all the command register bits are initialized to zero and the device is in VGA mode.												
BLANK*	I	98	Composite blank control input (TTL compatible). A logical zero drives the analog outputs to the blanking level, as specified in Figures 8 and 9. It is latched on the rising edge of GLCLK. When BLANK* is a logical zero, the pixel inputs are ignored.												
ENABLE (Composite Display Enable)	I	96	Composite display enable control input (TTL compatible). The state of this signal and BLANK* determines whether the analog outputs are blanked or contain cursor color, pixel, or overscan data. This signal is latched on the rising edge of GLCLK. If overscanning is not used, this pin should be tied to BLANK*. The following table lists the combinations of ENABLE and BLANK*:												
			<table border="1"> <thead> <tr> <th>ENABLE</th><th>BLANK*</th><th>Operation</th></tr> </thead> <tbody> <tr> <td>x</td><td>0</td><td>Video Blanking</td></tr> <tr> <td>0</td><td>1</td><td>Overscan Data</td></tr> <tr> <td>1</td><td>1</td><td>Cursor Color or Pixel Data</td></tr> </tbody> </table>	ENABLE	BLANK*	Operation	x	0	Video Blanking	0	1	Overscan Data	1	1	Cursor Color or Pixel Data
ENABLE	BLANK*	Operation													
x	0	Video Blanking													
0	1	Overscan Data													
1	1	Cursor Color or Pixel Data													
ODD/EVEN*	I	95	Odd/even field input (TTL compatible). This signal should be changed only during vertical blank. This input is used to ensure proper operation of the onboard cursor when interlaced operation (command bit CR23 = 1) is selected. When this signal is a logical zero, an even field is specified. When this signal is a logical one, an odd field is specified. This input is ignored if noninterlaced operation (command bit CR23 = 0) is selected.												
OSC, OSC*	I	131, 132	Pixel clock input (ECL/TTL compatible). This input is an ECL-compatible input, but a TTL clock may be used on either OSC or OSC* if selected by CR24 in Command Register 1 (CR34 = 0). It is recommended that all clock inputs be driven by a dedicated buffer to avoid reflection-induced jitter. In 1:1 mode DIVCLK = OSC or DIVCLK = OSC*.												
DIVCLK1	O	127	Frame buffer shift clock output (TTL compatible). The signal on this pin is equal to the selected pixel clock divided by 8, 4, 2, or 1, depending on the selection in the DIVCLK1 rate register. This output has low drive capability. DIVCLK1 and DIVCLK2 are opposite phases.												
DIVCLK2	O	128	Frame buffer shift clock output (TTL compatible). The signal on this pin is equal to the selected pixel clock divided by 8, 4, 2, or 1, depending on the selection in the DIVCLK2 rate register. This output has low drive capability. DIVCLK1 and DIVCLK2 are opposite phases.												
FIFO RESET*	I	14	A high value applied to this pin enables normal VideoCache™ FIFO operation. Although FIFO RESET* is level sensitive, a transition from high to low on this pin and should be kept low for at least 2 VLCLKs in order for the VideoCache™ FIFO to be properly reset.												
GLCLK	I	130	Graphics port input load clock (TTL compatible with hysteresis). The rising edge of this signal latches P7:0 (A-D), BLANK*, ENABLE, HSYNC*, and VSYNC*.												
VLCLK	I	13	Video port input load clock (TTL compatible with hysteresis). The rising edge of this signal latches P7:0 (E-H).												

## Pin Descriptions (continued)

Pin Name	I/O	Pin#	Description
P7:0 (A-H)	I	See Pin Diag	Pixel port inputs (TTL compatible). This port can be used in various modes as shown in Tables 10 and 11, for video and or graphics input.
VALID	I	15	Video port input pixel data valid signal (TTL compatible).
READY	O	12	Video port input pixel data ready signal (TTL compatible, low drive). This signal can be synchronously sampled using the rising edge of VLCLK. This signal changes only following a rising edge of VLCLK.
HFULL	O	11	VideoCache™ FIFO half-full or greater signal. (TTL compatible, low drive).
SEN	O	126	DIVCLK1 gating control signal (TTL compatible, low drive). It may be used to externally gate the DIVCLK1 output to generate a gated version of DIVCLK1. This signal changes only during DIVCLK1 low duration. The start time and duration of the pulse train may be programmed relative to the leading edge of C/HSYNC* and internally generated VSYNC.
VRESET*	O	2	Vertical reset signal (TTL compatible, low drive). This signal is generated to allow the asynchronous video data to know the start of each frame. This signal is synchronous to VLCLK.
MODE0	I/O	91	General purpose registered input/output (TTL compatible) set or read using CR45. Selection of input or output is made using CR37. Must be tied high with a 10K pullup resistor.
MODE1-3	I/O	29, 17, 19	Reserved for future expansion. Must be tied high with a 10K pullup resistor.
WR*	I	73	Write control input (TTL compatible with hysteresis). D0-D7 data is latched on the rising edge of WR*, and RS0-RS3 are latched on the falling edge of WR* during MPU write operations. RD* and WR* should not be asserted simultaneously.
RD*	I	74	Read control input (TTL compatible with hysteresis). To read data from the device, RD* must be a logical zero. RS0-RS3 are latched on the falling edge of RD* during MPU read operations. RD* and WR* should not be asserted simultaneously.
RS0-RS3	I	75-78	Register select inputs (TTL compatible). RS0-RS3 specify the type of read or write operation being performed, as specified in Tables 1 and 2.
D0-D7	I/O	83-90	Data bus (TTL compatible). Data is transferred into and out of the device over this 8-bit bidirectional data bus. D0 is the least significant bit.
SENSE*	O	71	Comparator sense output (CMOS compatible). This pin will be low if one or more of the IOR, IOG, and IOB analog output levels is above the internal comparator reference of 350 mV ± 50 mV.
IOR, IOG, IOB	A.O	55, 58, 61	Red, green, and blue current outputs. These high-impedance current sources can directly drive a doubly-terminated 75 Ω coaxial cable (see the PC Board Layout Considerations section for further information).

## Pin Descriptions (continued)

Pin Name	I/O	Pin#	Description																			
C/HSYNC*	I	97	Horizontal or composite sync control input (TTL compatible).																			
VSYNC*	I	99	Vertical sync control input (TTL compatible). This signal is pipelined to VSYNCOUT*.																			
HSYNCOUT*, VSYNCOUT*	O	48, 49	Pipeline delayed horizontal and vertical sync control signals.																			
FSADJUST	AJ	52	Full-scale adjust control. The IRE relationships in Figures 8 and 9 are maintained, regardless of the full-scale output current.  When an external or the internal voltage reference is used (see Figures 10 and 11 in the PC Board Layout Considerations section), a resistor (RSET) connected between this pin and GND controls the magnitude of the full-scale video signal. The relationship between RSET and the full-scale output current on each output is:  $RSET (\Omega) = K \cdot 1,000 \cdot VREF (V) / I_{out} (mA)$  K is defined in the table below. It is recommended that a 147 $\Omega$ RSET resistor be used for doubly-terminated 75 $\Omega$ loads (i.e., RS-343A applications). <table><tr><th rowspan="2">Setup</th><th colspan="2">Sync Enabled</th><th colspan="2">Sync Disabled</th></tr><tr><th>0 IRE</th><th>7.5 IRE</th><th>0 IRE</th><th>7.5 IRE</th></tr><tr><td>K (8-bit)</td><td>2.888</td><td>3.055</td><td>2.045</td><td>2.207</td></tr><tr><td>K (6-bit)</td><td>3.000</td><td>3.170</td><td>2.100</td><td>2.260</td></tr></table>	Setup	Sync Enabled		Sync Disabled		0 IRE	7.5 IRE	0 IRE	7.5 IRE	K (8-bit)	2.888	3.055	2.045	2.207	K (6-bit)	3.000	3.170	2.100	2.260
Setup	Sync Enabled		Sync Disabled																			
	0 IRE	7.5 IRE	0 IRE	7.5 IRE																		
K (8-bit)	2.888	3.055	2.045	2.207																		
K (6-bit)	3.000	3.170	2.100	2.260																		
VREF OUT	A.O	68	K values are subject to change upon completion of characterization.  Voltage reference output. This output provides a 1.2 V (typical) reference and may be connected directly to the VREFIN pin. If the on-chip reference is not used, this pin may be left floating. See Figures 10 and 11.																			
VREF IN	AJ	67	Voltage reference input. If an external voltage reference is used (Figure 11), it must supply this input with a 1.2 V (typical) reference. A 0.1 $\mu$ F ceramic capacitor must be used to decouple this input to GND, as shown in Figures 10 and 11. The decoupling capacitor must be as close to the device as possible to keep lead lengths to an absolute minimum. When the internal reference is used, this pin should not drive any external circuitry, other than the decoupling capacitor (Figure 10).																			
COMP	A.O	64	Compensation pin. A 0.1 $\mu$ F ceramic capacitor must be used to bypass this pin to the nearest VAA pin. The COMP capacitor must be as close as possible to the device to keep lead lengths to an absolute minimum. The compensation capacitor must be connected with short wide traces.																			
VAA	A.P	See Pin Diag	Analog power. All VAA pins must be connected to the same analog power plane.																			
GND	G	See Pin Diag	Analog ground. All GND pins must be connected to the same common ground plane.																			



**Note: All pins marked NC are reserved for future expansion and *MUST* be left floating.**

## PC Board Layout Considerations

### PC Board Considerations

The layout should be optimized for lowest noise on the Bt885 power and ground planes by providing good decoupling. The trace length between groups of VAA and GND pins should be as short as possible to minimize inductive ringing.

A well-designed power distribution network is critical to eliminating digital switching noise. The ground plane must provide a low-impedance return path for the digital circuits. A PC board with a minimum of four layers is recommended, with layers 1 (top) and 4 (bottom) for signals and layers 2 and 3 for ground and power.

### Component Placement

Components should be placed as close as possible to the associated CacheDAC™ pin. Whenever possible, components should be placed so traces can be connected point to point.

The optimum layout enables the Bt885 to be located as close as possible to the power supply connector and the video output connector.

### Ground Planes

For optimum performance, a common digital and analog ground plane is highly recommended.

### Power Planes

Separate digital and analog power planes are recommended. The digital power plane should provide power to all digital logic on the PC board, and the analog power plane should provide power to all Bt885 power pins, VREF circuitry, and COMP. There should be at least a 1/8-inch gap between the digital power plane and the analog power plane.

The analog power plane should be connected to the digital power plane (VCC) at a single point through a ferrite bead, as illustrated in Figures 10 and 11. This bead should be located within 3 inches of the Bt885. The bead provides resistance to switching currents, acting as a resistance at high frequencies. A low-resistance bead should be used, such as Ferroxcube 5659065-3B, TDK HF30ACB321611T, or TDK BF45-4001.

### Device Decoupling

For optimum performance, all capacitors should be located as close as possible to the device, and the shortest possible leads (consistent with reliable operation) should be used to reduce the lead inductance. Chip capacitors are recommended for minimum lead inductance. Radial lead ceramic capacitors may be substituted for chip capacitors and are better than axial lead capacitors for self-resonance. Values are chosen to have self-resonance above the pixel clock.

### Power Supply Decoupling

The best power supply decoupling performance is obtained with a 0.1  $\mu\text{F}$  ceramic capacitor in parallel with a 0.01  $\mu\text{F}$  chip capacitor decoupling each group of VAA pins to GND. The capacitors should be placed as close as possible to the device VAA and GND pins and connected with short, wide traces.

The 10  $\mu\text{F}$  capacitor shown in Figures 10 and 11 is for low-frequency power supply ripple; the 0.1  $\mu\text{F}$  capacitors are for high-frequency power supply noise rejection.

### COMP Decoupling

The COMP pin must be decoupled to VAA, typically with a 0.1  $\mu\text{F}$  ceramic capacitor and optional 15  $\Omega$  resistor. Low-frequency supply noise will require a larger value. The COMP capacitor must be as close as possible to the COMP and VAA pins. A surface-mount ceramic chip capacitor is preferred for minimal lead inductance. Lead inductance degrades the noise rejection of the circuit. Short, wide traces will also reduce lead inductance.

If the display has a ghosting problem, additional capacitance in parallel with the COMP capacitor may help.

### VREF Decoupling

A 0.1  $\mu\text{F}$  ceramic capacitor should be used to decouple this input to GND.

## PC Board Layout Considerations (continued)

### Digital Signal Interconnect

The digital inputs to the Bt885 should be isolated as much as possible from the analog outputs and other analog circuitry. Also, these input signals should not overlay the analog power plane or analog output signals.

Most of the noise on the analog outputs will be caused by excessive edge rates (less than 3 ns), overshoot, undershoot, and ringing on the digital inputs.

The digital edge rates should not be faster than necessary, as feedthrough noise is proportional to the digital edge rates. Lower-speed applications will benefit from using lower-speed logic (3–5 ns edge rates) to reduce data-related noise on the analog outputs.

Transmission lines will mismatch if the lines do not match the source and destination impedance. This will degrade signal fidelity if the line length reflection time is greater than one fourth the signal edge time (refer to Brooktree Application Notes AN-11 and AN-12). Line termination or line-length reduction is the solution. For example, logic edge rates of 2 ns require line lengths of less than 4 inches without use of termination. Ringing may be reduced by damping the line with a series resistor (30–300  $\Omega$ ). The RS-select inputs and RD\*/WR\* lines must be verified for proper levels with no ringing, undershoot, or overshoot. Ringing on these lines can cause improper operation.

Radiation of digital signals can also be picked up by the analog circuitry. This is prevented by reducing the digital edge rates (rise/fall time), minimizing ringing with damping resistors, and minimizing coupling through PC board capacitance by routing the digital signals at a 90 degree angle to any analog signals.

The clock driver and all other digital devices must be adequately decoupled to prevent noise generated by the digital devices from coupling into the analog circuitry.

### TTL Clock Interfacing

The Bt885 requires a pixel clock with monotonic clock edges for proper operation. Impedance mismatch on the pixel clock line will induce reflections on the pixel clock, which may cause erratic operation.

The Pixel Clock Pulse Width High Time and Pixel Clock Pulse Width Low Time minimum specifications (see the AC Characteristics section) must not be violated, or erratic operation can occur.

The pixel clock line must be terminated to prevent impedance mismatch. A series termination of 33–68  $\Omega$  placed at the pixel clock driver may be used, or a parallel termination may be used at the pixel clock input to

the CacheDAC™. A parallel termination of 220  $\Omega$  to VCC and 330  $\Omega$  to ground will provide a Thevenin equivalent of a 110  $\Omega$  termination, which is normally sufficient to absorb reflections. The series or parallel resistor values should be adjusted to provide the optimum clock signal fidelity.

### Differential Clock Interfacing

Termination requirements for differential ECL clock sources will vary depending on the particular clock generator used.

### MPU Control Signal Interfacing

The Bt885 uses the RD\*, WR\*, and RS lines to determine which MPU accesses will take place. Glitches or ringing on any of these lines may cause improper MPU operation. When a VGA controller with edge rate control is used on these lines, a series termination is not necessary. In non-VGA controller application or in applications where the MPU control signals are daisy chained, a series termination, pull-down resistors, or additional capacitance to ground should be used to prevent glitches that could cause improper MPU accesses.

### Analog Signal Interconnect

The Bt885 should be located as close as possible to the output connectors to minimize noise pickup and reflections caused by impedance mismatch.

The analog outputs are susceptible to crosstalk from digital lines; digital traces must not be routed under or adjacent to the analog output traces.

To maximize the high-frequency power supply rejection, the video output signals should not overlay the analog power plane.

For maximum performance, the analog video output impedance, cable impedance, and load impedance should be the same. The load resistor connection between the video outputs and GND should be as close as possible to the Bt885 to minimize reflections. Unused analog outputs should be connected to GND.

Analog output video edges exceeding the CRT monitor bandwidth can be reflected, producing cable-length dependent ghosts. Simple pulse filters can reduce high-frequency energy, reducing EMI and noise. The filter impedance must match the line impedance.

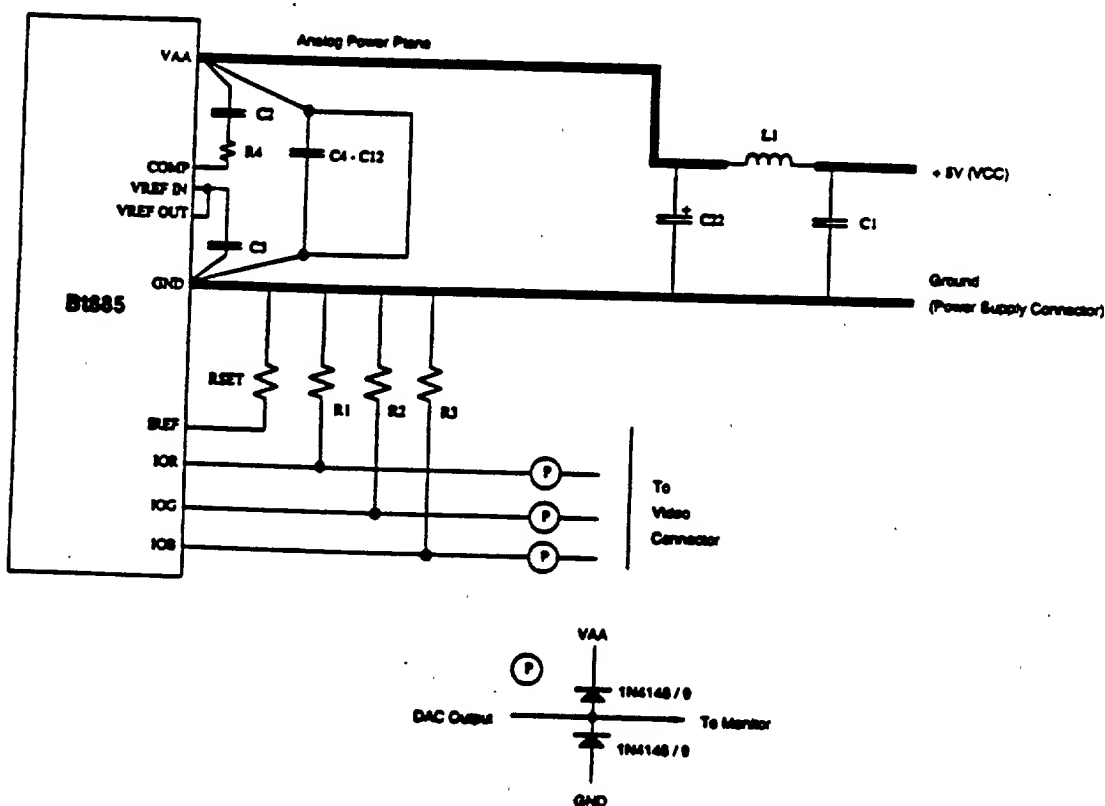
## PC Board Layout Considerations (continued)

## Analog Output Protection

The Bt885 analog output should be protected against high-energy discharges, such as those from monitor arc-over or from hot-switching AC-coupled monitors.

The diode protection circuit shown in Figures 10 and 11 can prevent latchup under severe discharge condi-

tions without adversely degrading analog transition times. The 1N4148/9 parts are low-capacitance, fast-switching diodes, which are also available in multiple-device packages (FSA250X or FSA270X) or surface-mountable pairs (BAV99 or MMBD7001).

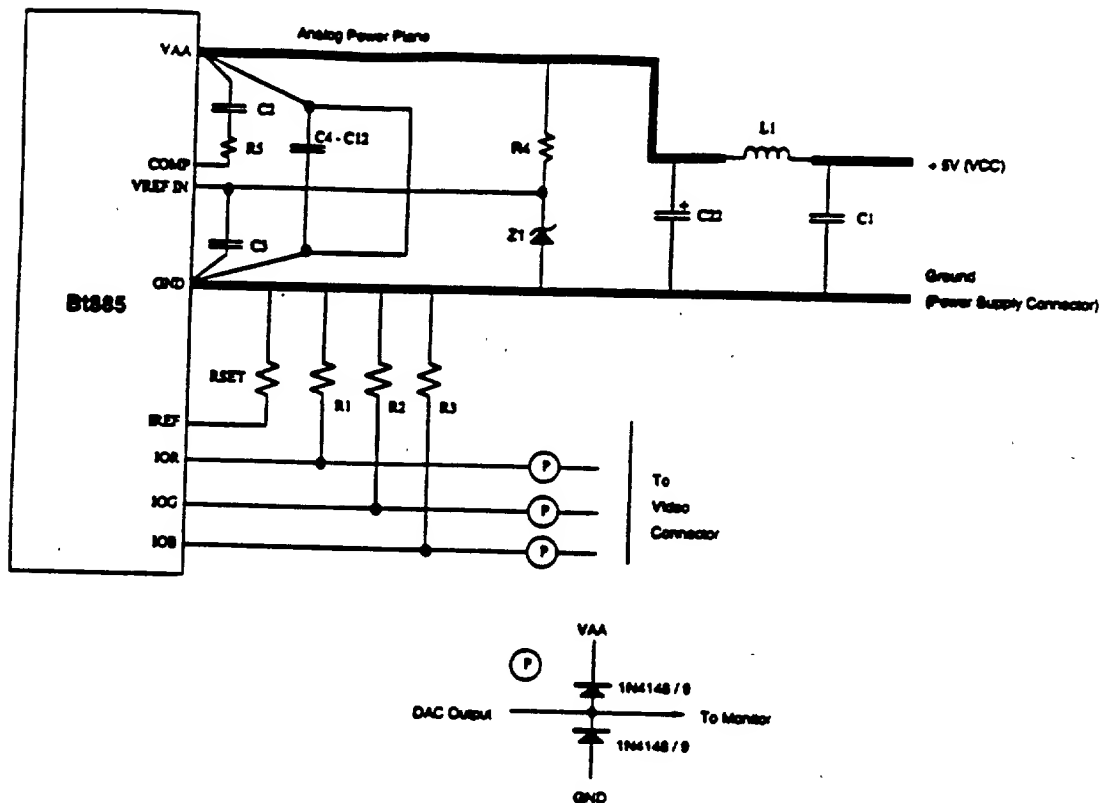


Location	Description	Vendor Part Number
C1-C12	0.1 $\mu$ F ceramic capacitor	Ene RPE11223U104M30V
C22	10 $\mu$ F capacitor	Mallory CSR13G106KM
L1	ferrite bead	TDK HF30ACB321611T
R1, R2, R3	75 $\Omega$ 1% metal film resistor	Dale CMF-55C
R4	15 $\Omega$ 1% metal film resistor	Dale CMF-55C
RSET	1% metal film resistor	Dale CMF-55C

Figure 10. Typical Connection Diagram and Parts List (Internal Voltage Reference).



PC Board Layout Considerations (continued)



Location	Description	Vendor Part Number
C1-C12	0.1 $\mu$ F ceramic capacitor	Eric RPE11223U104M50V
C22	10 $\mu$ F capacitor	Mallory CSR13G106KM
L1	ferrite bead	TDK HF30ACB321611T
R1, R2, R3	75 $\Omega$ 1% metal film resistor	Dale CMF-55C
R4	1 K $\Omega$ 5% metal film resistor	Dale CMF-55C
R5	15 $\Omega$ 1% metal film resistor	Dale CMF-55C
RSET	1% metal film resistor	Dale CMF-55C
Z1	1.2 V voltage reference	National Semiconductor LM385BZ-1.2

Figure 11. Typical Connection Diagram and Parts List (External Voltage Reference).

## Application Information

### *Using Multiple Devices*

When multiple Bt885s are used, each Bt885 should have its own power plane and ferrite bead. If the internal reference is used, each Bt885 should use its own internal reference.

Although the multiple Bt885s may be driven by a common external voltage/current reference, higher performance may be obtained if each CacheDAC™ uses its own reference. This will reduce the amount of color channel crosstalk and color palette interaction.

Each Bt885 must still have its own RSET resistor, analog output termination resistors, power supply bypass capacitors, COMP capacitor, and reference capacitors.

### *ESD and Latchup Considerations*

Correct ESD-sensitive handling procedures are required to prevent device damage, which can produce symptoms of catastrophic failure or erratic device behavior with leaky inputs.

All logic inputs should be held low until power to the device has settled to the specified tolerance. DAC power decoupling networks with large time constants should be avoided. They could delay VAA power to the device. Ferrite beads must be used only for analog power VAA decoupling. Inductors and regulators cause a time constant delay that induces latchup.

Latchup can be prevented by ensuring that all VAA and GND pins are at the same potential and that the VAA supply voltage is applied before the signal pin voltages. The correct power-up sequence ensures that any signal pin voltage will never exceed the power supply voltage by more than +0.5 V.

### *Sleep Operation*

When the internal or external voltage reference is used, the DACs will be turned off during sleep mode.

When an external voltage reference is used, some internal circuitry will still be powered during the sleep mode. This unnecessary current drain can be disabled by turning off the external voltage reference during power-down mode.

## Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Power Supply	VAA	4.75	5.00	5.25	V
Ambient Operating Temperature	TA	0		+70	°C
Output Load	RL		37.5		$\Omega$
Voltage Reference Configuration Reference Voltage	VREF	1.1112	1.235	1.359	V

## Absolute Maximum Ratings

Parameter	Symbol	Min	Typ	Max	Units
VAA (measured to GND)				7.0	V
Voltage on Any Signal Pin (Note 1)		GND-0.5		VAA + 0.5	V
Analog Output Short Circuit Duration to Any Power Supply or Common	ISC		indefinite		
Storage Temperature	TS	-65		+150	°C
Junction Temperature	TJ			+150	°C
Vapor Phase Soldering (1 minute)	TVSOL			220	°C

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

*Note 1:* This device employs high-impedance CMOS devices on all signal pins. It should be handled as an ESD-sensitive device. Voltage on any signal pin that exceeds the power supply voltage by more than +0.5 V can cause destructive latchup.

## DC Characteristics

Parameter	Symbol	Min	Typ	Max	Units
Resolution (each DAC)		8	8	8	
Accuracy (each DAC)					
Integral Linearity Error	IL			±1	LSB
Differential Linearity Error	DL			±1	LSB
Gray-Scale Error			Guaranteed	±5	% Gray
Monotonicity					Scale
Coding					Binary
Digital Inputs					
Input High Voltage	V <sub>IH</sub>	2.0		V <sub>AA</sub> + 0.5	V
Input Low Voltage	V <sub>IL</sub>	GND-0.5		0.8	V
Input High Current (V <sub>in</sub> = 2.4 V)	I <sub>IH</sub>			TBD	μA
Input Low Current (V <sub>in</sub> = 0.4 V)	I <sub>IL</sub>			TBD	μA
Input Capacitance	C <sub>IN</sub>			7	pF
(v = 1 MHz, V <sub>in</sub> = 2.4 V)					
Hysteresis			0.3		V
OSC/OSC* ECL					
Differential Inputs	Δ V <sub>in</sub>	0.6			V
Input High Voltage	V <sub>IH</sub>	V <sub>CC</sub> -1.1		V <sub>CC</sub> -0.8	V
Input Low Voltage	V <sub>IL</sub>	V <sub>CC</sub> -2		V <sub>CC</sub> -1.5	V
Digital Outputs					
Output High Voltage	V <sub>OH</sub>	2.4			V
(I <sub>OH</sub> = -400 μA)					
Output Low Voltage	V <sub>OL</sub>			0.4	V
(I <sub>OL</sub> = 3.2 mA)					
Three-State Current	I <sub>OZ</sub>			10	μA
Output Capacitance	C <sub>DOUT</sub>			7	pF
Load Capacitance	C <sub>L</sub>			10	pF

See test conditions on next page.

## DC Characteristics (continued)

Parameter	Symbol	Min	Typ	Max	Units
Analog Outputs					
Gray-Scale Current Range				20	mA
Output Current (Standard RS-343A)					
White Level Relative to Black		15.86	17.62	18.5	mA
Black Level Relative to Blank					
SETUP = 7.5 IRE		0.95	1.44	1.90	mA
SETUP = 0 IRE		0	5	50	μA
Blank Level		6.29	7.62	8.96	mA
Sync Level		0	5	50	μA
LSB Size			69.1		μA
DAC-to-DAC Matching			2	5	%
Output Compliance	VOC	-1.0		+1.5	V
Output Impedance	RAOUT		10		kΩ
Output Capacitance	CAOUT			30	pF
(f = 1 MHz, IOUT = 0 mA)					
Onboard VREF (Note 1)	VREFOUT	TBD	TBD	TBD	V
Voltage Reference Input Current	IVR IN		tbd	tbd	mA
Power Supply Rejection Ratio	PSRR			0.5	% / % ΔVAA
(COMP = 0.1 μF, f = 1 kHz)					

Test conditions to generate RS-343A standard video signals (unless otherwise specified): "Recommended Operating Conditions" using external voltage reference with SETUP = 7.5 IRE, RSET = 147 Ω, VREF = 1.235 V. As the above parameters are guaranteed over the full temperature range, temperature coefficients are not specified or required.

When the internal voltage reference is used, RSET may require adjustment to meet these limits. Also, the "gray-scale" output current (white level relative to black) will have a typical tolerance of ±10% rather than the ±5% specified above.

When the device is in the 6-bit mode, the output levels are approximately 1.5% lower than these values.

Note 1: Onboard VREF numbers subject to change upon completion of characterization.

## AC Characteristics

Parameter	Symbol	110 MHz Devices			Units
		Min	Typ	Max	
OSC, OSC* All Mux Rates	Fmax			110	MHz
RS0-RS3 Setup Time	1	10			ns
RS0-RS3 Hold Time	2	10			ns
RD* Asserted to D0-D7 Driven	3	2			ns
RD* Asserted to D0-D7 Valid	4			40	ns
RD* Negated to D0-D7 3-States	5			20	ns
Read D0-D7 Hold Time	6				ns
Write D0-D7 Setup Time	7	10			ns
Write D0-D7 Hold Time	8	10			ns
RD*, WR* Pulse Width Low	9	50			ns
RD*, WR* Pulse Width High	10	6* pixel clock periods			ns
GLCLK Rates	Gmax				
8:1 Multiplexing				13.75	MHz
4:1 Multiplexing				27.5	MHz
2:1 Multiplexing				55	MHz
1:1 Multiplexing				90	MHz
VLCLK Rate	Vmax			85	MHz
DIVCLK1, DIVCLK2 Rates	Dmax			55	MHz
OSC, OSC* Cycle Time (Note 1)	11	18.18			ns
All Mux Rates					
OSC, OSC* Pulse Width High	12	tbd			ns
All Mux Rates					
OSC, OSC* Pulse Width Low	13	tbd			ns
All Mux Rates					
Duty Cycle of Selected Pixel Clock		45		55	%
When Clock Doubler Enabled					
GLCLK Cycle Time	14				
8:1 Multiplexing		72.72			ns
4:1 Multiplexing		36.36			ns
2:1 Multiplexing		18.18			ns
1:1 Multiplexing		11.11			ns
GLCLK Pulse Width High	15				
8:1 Multiplexing		4			ns
4:1 Multiplexing		4			ns
2:1 Multiplexing		4			ns
1:1 Multiplexing		4			ns
GLCLK Pulse Width Low	16				
8:1 Multiplexing		4			ns
4:1 Multiplexing		4			ns
2:1 Multiplexing		4			ns
1:1 Multiplexing		4			ns

Test conditions at end of this section.

## AC Characteristics (continued)

Parameter	Symbol	110 MHz Devices			Units
		Min	Typ	Max	
VLCLK Cycle Time	17	11.76			ns
VLCLK Pulse Width High	18	4			ns
VLCLK Pulse Width Low	19	4			ns
DIVCLK1, DIVCLK2 Cycle Time	20	14.81			ns
DIVCLK1, DIVCLK2 Duty Cycle	21	40		60	%
Graphics Data Setup to GLCLK	22	3			ns
Graphics Data Hold from GLCLK	23	1			ns
Data Setup to GLCLK ENABLE, BLANK*, CHSYNC*, VSYNC*	24	3			ns
Data Hold to GLCLK ENABLE, BLANK*, CHSYNC*, VSYNC*	25	1			ns
Video Data Setup to VLCLK	26				ns
Video Data Hold from VLCLK	27				ns
VALID Setup to VLCLK	28	3			ns
VALID Hold from VLCLK	29	1			ns
VLCLK to READY Valid	30			7	ns
DIVCLK1 to SEN Valid	31			3	ns
FIFO Reset Pulse Width		2 * VLCLK periods			ns
Analog Output Delay	32			30	ns
Analog Output Rise/Fall Time	33		3		ns
Analog Output Settling Time (Note 2)	34		13		ns
Clock and Data Feedthrough (Note 2)			-30		dB
Glitch Impulse (Note 2)			75		pV - sec
SENSE* Output Delay	35		1		µs
DAC-to-DAC Crosstalk			-23		dB
Analog Output Skew				2	ns
VAA Supply Current	IAA				mA
Normal Operation			tbd	tbd	mA
"Sleep" Mode (Note 3)			tbd	tbd	mA

Test conditions at end of this section.

## AC Characteristics (continued)

Pipeline Delay	
Graphics 1:1/No Video	3 LCLKS + 16 PCLKS
MUX Graphics/No Video	(8 LCLKS + 16 PCLKS) ± 2 LCLKS
Graphics 1:1/Video	27 LCLKS + 16 PCLKS
MUX Graphics/Video	(32 LCLKS + 16 PCLKS) ± 2 LCLKS
The number of LCLKS will have to be multiplied by the respective MUX rates to get the proper number of pipeline delays. (i.e., PCLK = Pixel Clock Rate LCLK = MUX Clock Rate the pipeline delay in 2:1 MUX Graphics/No video, measured in PCLKs = 32 PCLKS ± 4 PCLKs.	

Test conditions (unless otherwise specified): "Recommended Operating Conditions" using external voltage reference with SETUP = 7.5 IRE, VREF = 1.235 V, RSET = 147Ω TTL input values are 0–3 V, with input rise/fall times ≤ 3 ns, measured between the 10% and 90% points. Timing reference points at 50% for inputs and outputs. Analog output load ≤ 10 pF; SENSE\* and D0–D7 output load ≤ 50 pF. DIVCLK1, DIVCLK2 output load = 50 pF. As the above parameters are guaranteed over the full temperature range, temperature coefficients are not specified or required. Timing waveforms are shown in Figures 12–14.

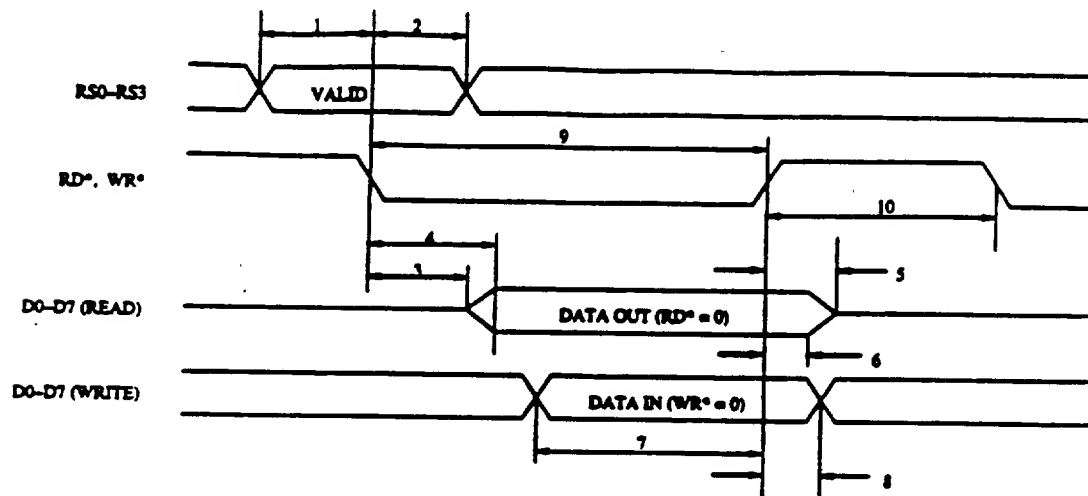
Note 1: OSC and OSC\* cycle times assume the use of the 2Xclock Multiplier.

Note 2: Numbers guaranteed by design.

Note 3: External voltage reference is disabled during sleep mode, all inputs are low, and clock is running.

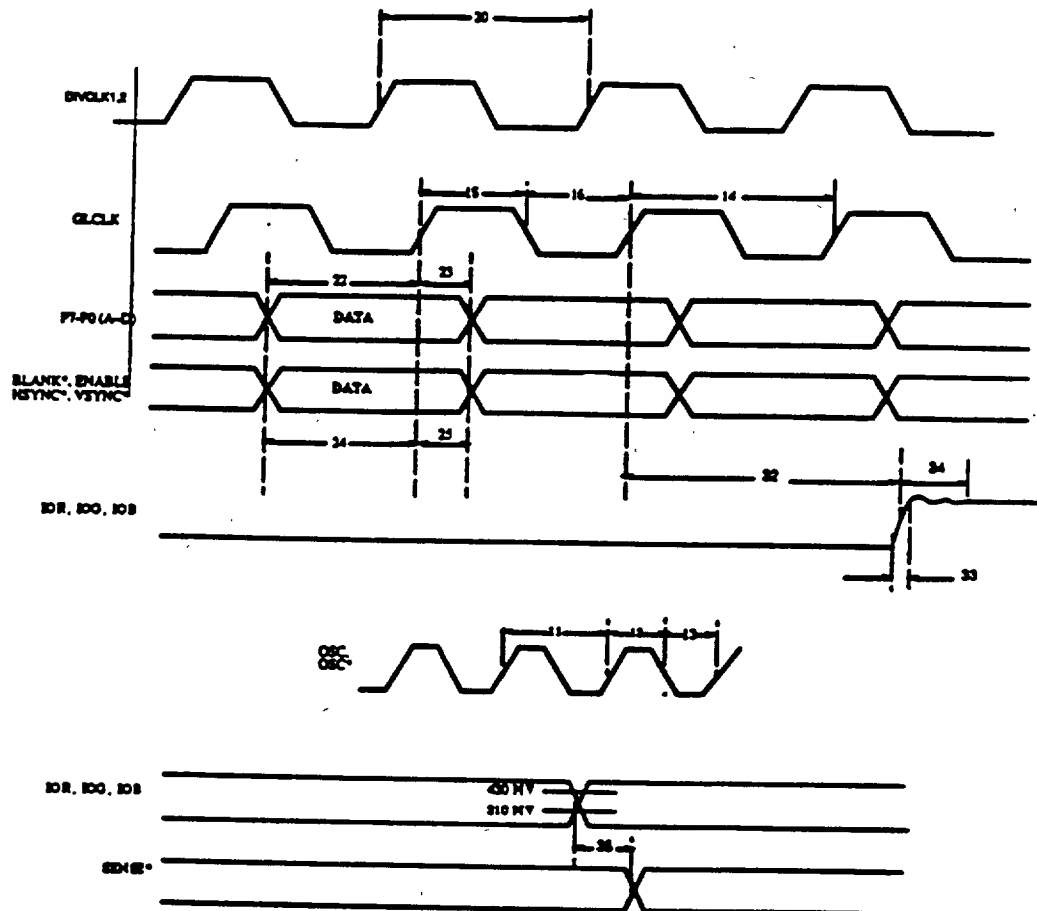


# Timing Waveforms



- Note 1:* Output delay measured from the 50% point of the rising edge of CLOCK to the 50% point of full-scale transition.
- Note 2:* Settling time measured from the 50% point of full-scale transition to the output remaining within  $\pm 1$  LSB.
- Note 3:* Output rise/fall time measured between the 10% and 90% points of full-scale transition.

**Figure 12. MPU Read/Write Timing.**



- Note 1:** Output delay measured from the 50% point of the rising edge of CLOCK to the 50% point of full-scale transition.
- Note 2:** Settling time measured from the 50% point of full-scale transition to the output remaining within  $\pm 1$  LSB.
- Note 3:** Output rise/fall time measured between the 10% and 90% points of full-scale transition.

**Figure 13. Graphics Input/Output Timing.**

Timing Waveforms (continued)

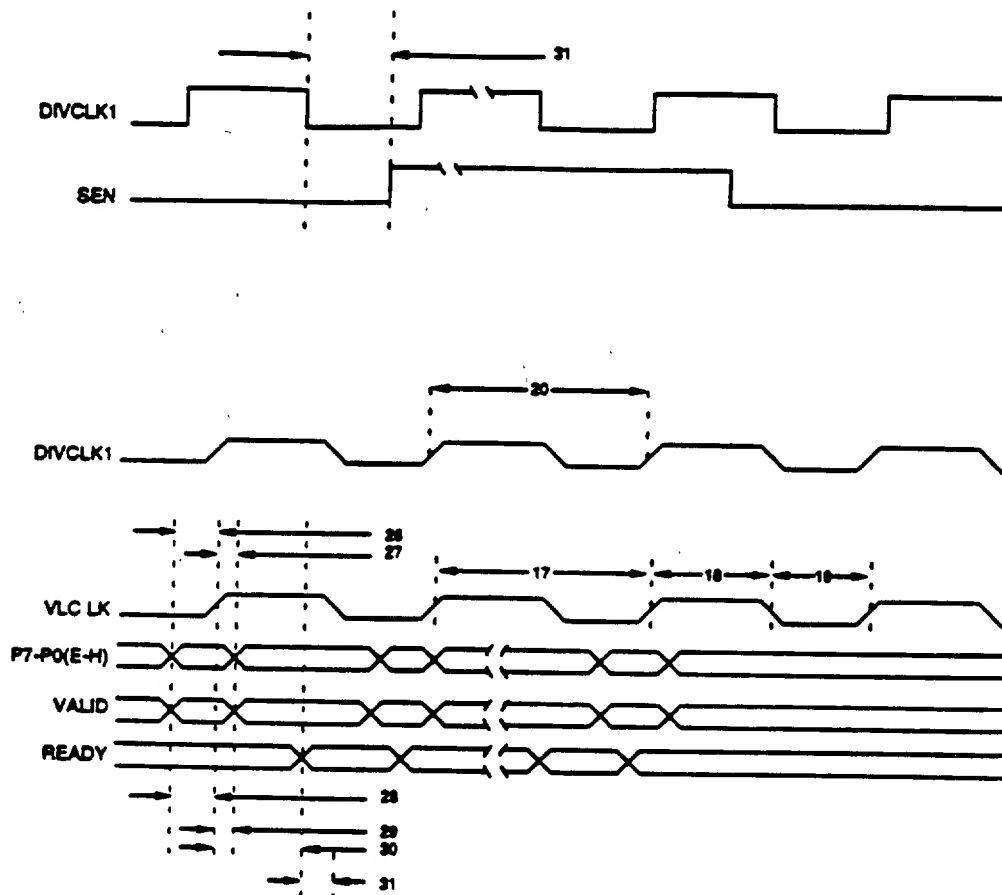


Figure 14. Video Input/Output Timing.

## Ordering Information

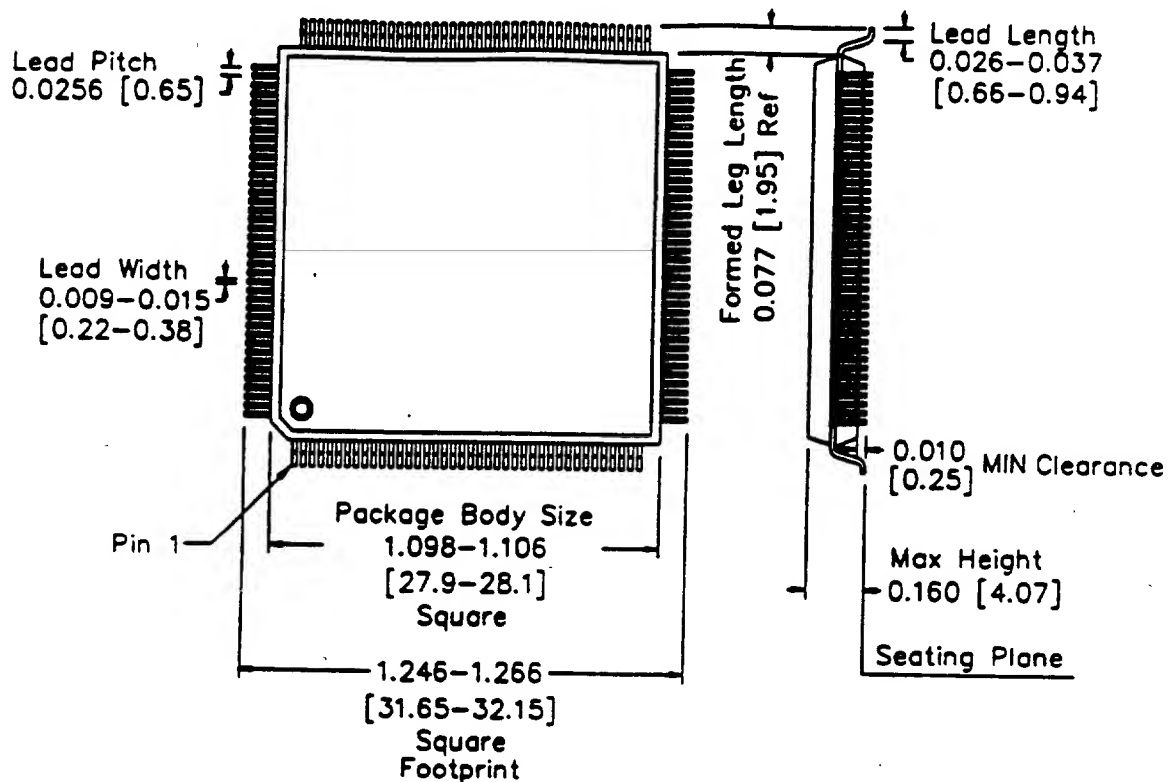
Model Number	Speed	Package	Ambient Temperature Range
Bt885KHF110	110 MHz	160-pin Plastic Quad Flatpack	0° to +70°C

## Revision History

**Datasheet  
Revision**
**Changes From Previous Revision**

- B** Initial Release
- C** Pinout change. Pin 160 was changed from GND to RPTLINE, Pin 2 was changed from GND to VRESET\*. Pins 1, 39, 40, 41, 42, 79, 80, 81, 82, 119, 120, 121, 122 were changed from GND to NC.  
Vertical scaling support added. Both DIVCLK1 and DIVCLK2 and all internal clocks are derived from OSC or OSC\* clock inputs.  
DIVCLK1 and DIVCLK2 outputs are opposite phases.
- D** 135 MHz speed grade removed. Pinout change.  
Pin 160 was changed from RPTLINE to NC.  
Vertical scaling deleted.

## Package Drawing—160-pin Plastic Quad Flatpack (PQFP)



Notes: Unless otherwise specified:

1. Dimensions are in inches [millimeters]. Millimeters are the controlling dimension.
2. Package body size does not include mold protrusion or mismatch.
3. PCB pad layout suggestions:
  - a. Pad size: 0.100 x 0.012 (2.54 x 0.30).
  - b. Lead pitch (millimeters): Use 0.65 center-to-center spacing.
  - c. Lead pitch (inches): If the PCB layout system to be used can handle fractional mils, use 0.0256 center-to-center spacing. If not, use a combination of 0.025(A) and 0.026(B) inch spacings in groups of five ("ABABA" repeated) to approximate the exact spacing as closely as possible. For example, "ABABA" "ABABA" and so forth.

# Brooktree®

Brooktree Corporation  
9868 Scranton Road  
San Diego, CA 92121-3707  
(619) 452-7580  
(800) 2-BT-APPS  
TLX: 383 596  
FAX: (619) 452-1249  
L885001 Rev. D

Information furnished by Brooktree Corporation is believed to be accurate and reliable. However, no responsibility is assumed for its use, nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Brooktree Corporation.

Copyright © 1994.  
Brooktree Corporation.

Specifications are subject to change without notice.

## CAUTION



ESD-sensitive device.  
Permanent damage may occur on unconnected devices subjected to high-energy electrostatic fields. Unused devices must be stored in conductive foam or shunts.

Do not insert this device into powered sockets.

Remove power before insertion or removal.



PRINTED IN THE USA  
Print date: 02/15/94

printed on recycled paper

AT1020014

## Advance

This document contains information on a product under development. The parametric information contains target parameters that are subject to change.

EXHIBIT

RX-238

## Distinguishing Features

- 135 and 110 MHz Pipelined Operation
- VGA Compatible
- Mixed Video and Graphics
- 64-bit Graphics/Video Pixel Ports
- YCrCb-to-RGB Conversion
- YCrCb 4:2:2 and 2:1:1 Interpolation
- Uses Brooktree's VideoCache™ Technology
- Horizontal Video Up-Scaling
- 64 x 64 x 2 Cursor
- VRAM Shift Clock Support
- Enables DRAM-Based Motion Video Systems
- Programmable Video Extents
- Programmable Color Keying

- Three 256 x 8 Color Palette RAMs
- Simplifies Integration of Video into Microsoft Windows™
- 3 x 24 Cursor Color Palette
- Standard MPU Interface
- Power-Down Mode
- Directly Implements Brooktree's VideoCache™ Connector
- 160 PQFP

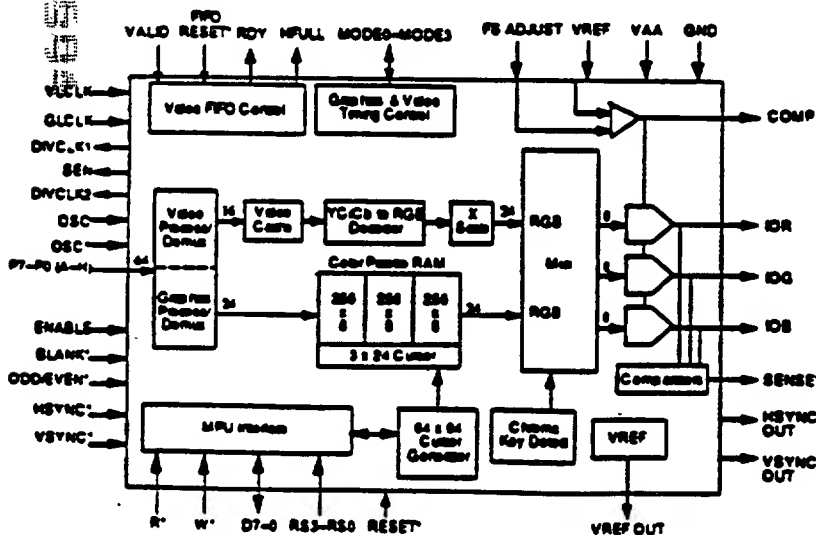
## Applications

- Video Decompression Acceleration
- Multimedia Workstations
- High-Resolution Graphics
- Desktop Video

## Related Products

- Bt812 Video Decoder
- Bt858 Video Encoder

## Functional Block Diagram



Brooktree Corporation • 9950 Barnes Canyon Rd. • San Diego, CA 92121  
(619) 452-7580 • (800) 2BT-APPS • TLX: 383 596 • FAX: (619) 452-1249  
L885001 Rev. B

# Bt885

## 135 MHz Monolithic CMOS Video CacheDACTM

### Product Description

The Bt885 is designed specifically for dual or unified frame buffer multimedia subsystems. A dedicated video port accepts a CCIR601 YCrCb data stream and allows on-screen switching on a pixel-by-pixel basis. Mixing occurs within programmable video extents based on a flexible color key mechanism. Bt885 is intended to replace multiple RAMDAC™-based multimedia subsystems. The Bt885 register set is VGA compatible.

The Bt885 can accelerate video decompression and work with the Bt812 decoder chip using programmable interpolation to pixel multiply by 1, 2, or 4 for CCIR601 4:2:2, 2:1:1, and 1:0.5:0.5 formats. This allows the video data to mix with the graphics data at the same rate.

Brooktree's 800-byte VideoCache™ FIFO enables asynchronous delivery of graphics and video, easing system bandwidth requirements for video transfer, and allowing efficient use of system memory. Non-integer scaling permits arbitrary video window sizing.

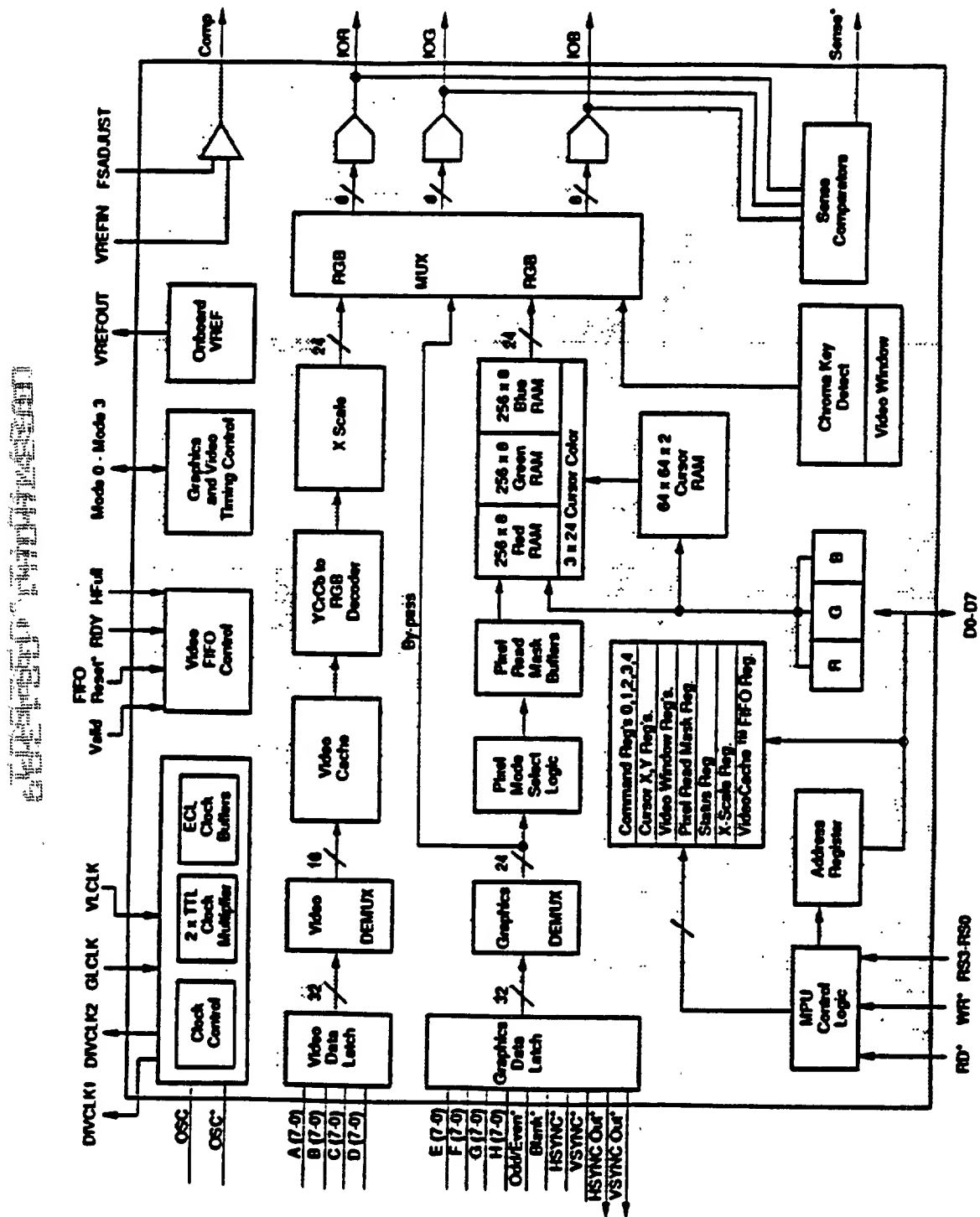
The 64 x 64 x 2 bit cursor has its own palette and has priority over the video or graphics. The cursor operates in three modes: Microsoft Windows™, three color, and X Windows.

The Bt885 supports independent 32-bit graphics and 32-bit video pixel ports and is compatible with both VRAM- and DRAM-based video subsystems.

The Bt885 generates RS-343A-compatible video signals into a doubly-terminated 75 Ω load.

Brooktree®

ATI020015



**Figure 1. Bt885 Detailed Block Diagram.**



## Circuit Description (continued)

**MPU Interface**

As illustrated in the detailed block diagram, a standard MPU bus interface is supported, allowing the MPU direct access to the color palette RAM. MPU data is transferred into and out of the CacheDAC™ through the D0-D7 data pins. The read/write timing is controlled by the RD\* and WR\* inputs.

The RS0-RS3 select inputs specify which control register the MPU is accessing, as shown in Table 1. The 8-bit address register is used to address the color palette RAM, eliminating the requirement for external address multiplexers. D0 corresponds to ADDR0 and is the least significant bit.

**Hardware Reset Condition**

On reset, Bt885 is configured for standard VGA compatibility as follows:

- 8 bits per pixel graphics, 1:1 MUX.
- 6-bit DAC resolution.
- Pixel mask register set to 0xFF.
- Video modes disabled.
- All control registers set for VGA compatibility.
- Graphic pipelines are reset.

**Writing Color Palette RAM Data**

To write color data, the MPU writes the address register (RAM write mode) with the address of the color palette RAM location to be modified. The MPU performs three successive write cycles (8 bits each of red, green, and blue), using RS0-RS3 to select the color palette RAM. After the blue write cycle, the 3 bytes of color information are concatenated into a 24-bit word and written to the location specified by the address register. The address register then increments to the next location, which the MPU may modify by writing another sequence of red, green, and blue data. A block of color values in consecutive locations may be written to by writing the start address and performing continuous RGB write cycles until the entire block has been written. Refer to the Timing Waveforms for further information.

**Reading Color Palette RAM Data**

To read color palette RAM data, the MPU loads the address register (RAM read mode) with the address of the color palette RAM location to be read. The contents of the color palette RAM at the specified address are

copied into the red, green, or blue (RGB) registers and the address register is incremented to the next RAM location. The MPU performs three successive read cycles (8 bits each of red, green, and blue), using RS0-RS3 to select the color palette RAM. Following the blue read cycle, the contents of the color palette RAM at the address specified by the address register are copied into the RGB registers, and the address register increments again. A block of color values in consecutive locations may be read by writing the start address and performing continuous RGB read cycles until the entire block has been read.

RS3-RS0	Access	Addressed by MPU
0000	R/W	address register; palette/cursor RAM write
0001	R/W	6/8-bit color palette data
0010	R/W	pixel mask register
0011	R/W	address register; palette/cursor RAM read
0100	R/W	address register; cursor/overscan color write
0101	R/W	cursor overscan and color data
0110	R/W	command register 0
0111	R/W	address register; cursor/overscan color read
1000	R/W	command register 1
1001	R/W	command register 2
1010	R/W	extended address read/write register
1011	R/W	cursor RAM array data
1100	R/W	cursor x-low register
1101	R/W	cursor x-high register
1110	R/W	cursor y-low register
1111	R/W	cursor y-high register

Table 1. Control Input Truth Table  
(RS3 = MSB, RS0 = LSB).

**Writing Cursor and Overscan Color Data**

To write cursor or overscan color data, the MPU writes the address register (cursor color write mode) with the address of the cursor or overscan color location to be modified. The MPU performs three successive write cycles (8 bits each of red, green, and blue), using RS0-RS3 to select the cursor color registers. After the blue write cycle, the 3 bytes of red, green, and blue color information are concatenated into a 24-bit word and written to the cursor or overscan color location specified

## Circuit Description (continued)

by the address register. The address register then increments to the next location, which the MPU may modify by writing another sequence of red, green, and blue data. A block of color values in consecutive locations may be written to by writing the start address and performing continuous RGB write cycles until the entire block has been written.

### Reading Cursor Color Data

To read cursor color data, the MPU loads the address register (cursor color read mode) with the address of the cursor color location to be read. The contents of the cursor color register at the specified address are copied into the RGB registers, and the address register is incremented to the next cursor color location. The MPU performs three successive read cycles (8 bits each of red, green, and blue), using RS0–RS3 to select the cursor color registers. Following the blue read cycle, the contents of the cursor color location at the address specified by the address register are copied into the RGB registers, and the address register again increments. A block of color values in consecutive locations may be read by writing the start address and performing continuous R, G, B read cycles until the entire block has been read.

### Extended Register Mechanism

An extended register set is used to accommodate all features of the Bt885. Since there are only four register select lines (and all 16 combinations have already been used), the extended registers must be accessed indirectly.

For example, Command Register 3 is accessed with the following sequence of operations:

1. Set RS3–RS0 = 0000, Address Register.
2. Write Address Register to 0x02
3. Set RS3–RS0 = 1010 (Extended Address Register).
4. Read or Write Command Register 3.

### Writing Color Key Color Data

To write the color key color data value, the MPU selects the color key data RGB register using the extended register. It then performs a write cycle setting RS3–RS0 to 1010 (Status Register). This process is repeated for each color component. The color key color register is only updated after the blue value is written.

### Reading Color Key Color Data

To read the color key color data value, the MPU selects the color key data RGB register using the extended register mechanism, then performs a read cycle setting RS3–RS0 to 1010 (Status Register).

### Writing Color Key Mask Data

To write the color key mask data value, the MPU selects the color key mask RGB register using the extended register mechanism. It then performs a write cycle setting RS3–RS0 to 1010 (Status Register). This process is repeated for each color component. The color key mask register is only updated after the blue value is written.

### Reading Color Key Mask Data

To read the color key color mask value, the MPU selects the color key data RGB register using the extended register mechanism outlined below, then performs a read cycle setting RS3–RS0 to 1010 (Status Register).

### Additional Information

When the color palette RAM is accessed, the address register resets to 0x00 following a blue read or write cycle to RAM location 0xFF.

The MPU interface operates asynchronously to the pixel clock. Data transfers between the color palette RAM and the color registers (R, G, and B in the functional block diagram) are synchronized by internal logic, and occur in the period between MPU accesses. To reduce noticeable sparkling on the CRT screen during MPU access to the color palette RAMs, internal logic maintains the previous output color data on the analog outputs while the transfer between RGB registers and lookup table RAMs occurs.

To keep track of the red, green, and blue read/write cycles, the address register has two additional bits (ADDRa and ADDRb) that count modulo three. They are reset to zero when the MPU writes to the address register and are not reset to zero when the MPU reads the address register. The MPU does not have access to these bits. The MPU may read the address register at any time without modifying its contents or the existing read/write mode.

## Circuit Description (continued)

**Accessing the Cursor RAM Array**

The 64 x 64 x 2 cursor RAM is accessed in a planar format. Bits CR31 and CR30 in Command Register 3 become the load inputs to the 2 MSBs of a 10-bit address counter; therefore, these bits must be written in Command Register 3 before the lower 8 bits are written to the address counter through the MPU port. In the planar format, only nine address bits are used. The tenth bit determines which plane (0 or 1) data of the cursor RAM array is accessed. A single address presented to the cursor RAM accesses 8-bit locations in plane 0 or 1, depending on the state of address bit 9.

After each access in the planar format, the address increments. The MPU uses ADDR, a 10-bit binary address counter, to access the cursor RAM array. The address counter is the same 8-bit binary counter used for RGB autoincrementing with CR31 and CR30 as its extended MSBs. Any write to the address counter after cursor autoincrementing has been initiated resets the cursor autoincrementing logic until cursor RAM array has again been accessed. Cursor autoincrementing will then begin from the address written. A read from the address counter does not reset the cursor, autoincrementing logic. The color palette RAM and the cursor RAM share the same external address register, and MPU addressing for this and all other registers is determined by the external register select lines RS3-RS0 (see Table 2).

**6-Bit/8-Bit Operation**

The command bit CR01 specifies whether the MPU is reading and writing 8 bits or 6 bits of color information each cycle. For an 8-bit operation, D0 is the LSB and D7 is the MSB of color data. For a 6-bit operation, color data is contained on the lower 6 bits of the data bus, with D0 as the LSB and D5 as the MSB of color data. When the MPU is writing color data, D6 and D7 are ignored. During color read cycles, D6 and D7 are a logical zero.

Accessing the cursor RAM array does not depend on the resolution of the DACs. When Bt885 is in the 6-bit mode, the 6-bit DAC values are left justified within an 8-bit field and the two LSBs are set to zero. Therefore, Bt885's full-scale output current will be about 1.5 percent lower than while it is in the 8-bit mode.

**Power-Down Mode**

The Bt885 incorporates a power-down capability, controlled by command bit CR00. While command bit CR00 is a logical zero, the Bt885 functions normally.

While command bit CR00 is a logical one, the DACs, cursor circuitry, video FIFO, and power to the RAM are turned off. The RAM still retains the data. Also, the RAM may be read or written to by the MPU as long as the pixel clock is running. The RAM automatically powers up during MPU read/write cycles and shuts down when the MPU access is completed. The DACs output no current, and the four command registers may still be written to or read by the MPU. The output DACs require about one second to turn off (sleep mode) or turn on (normal), depending on the compensation capacitor used (see the Video Output Waveforms section for further information). The DACs will be turned off during sleep mode only if a voltage reference (internal or external) is used.

When an external voltage reference is used, external circuitry should turn off the voltage reference ( $V_{REF} = 0$  V) to further reduce power consumption caused by biasing of portions of the internal voltage reference.

**Pixel Clock Selection**

OSC and OSC\* provide the source for the Bt885 internal pixel clock. In general, graphic pixel data is latched by GLCLK. Bit CR24 selects whether the OSC or OSC\* pin is used. A clock doubler can be enabled on the selected input by setting CR33 = 1. The OSC\* and OSC inputs can be used together as differential ECL inputs for the external clock by setting CR34 = 1. If a differential ECL input mode is used (CR34 = 1), then the state of CR33 and CR24 are ignored.

It is also possible to internally route the DIVCLK2 output to the latches connected to GLCLK by setting CR36 = 1. GLCLK will be ignored in this mode.

DIVCLK1 and DIVCLK2 are output on the basis of the OSC and OSC\* inputs as described unless they are disabled by setting CR32 = 0 (DIVCLK1 disable) or CR35 = 0 (DIVCLK2 disable). If the clock doubler is used (CR33 = 1), then both the DIVCLK1 and DIVCLK2 dividers must be set to a value of 2 or greater. For all graphics 1:1 modes, GLCLK is used as the pixel clock and the OSC and OSCB inputs are ignored.

## Circuit Description (continued)

CR31 (bit A9 of ADDR)	ADDR 0-7 (counts binary)	ADDR a,b (counts modulo 3)	RS3	RS2	RS1	RS0	Addressed by MPU
N/A	0x00-0xFF	00	0	0	0	1	Color palette RAM (Red Component)
		01	0	0	0	1	Color palette RAM (Green Component)
		10	0	0	0	1	Color palette RAM (Blue Component)
N/A	0x00	00	0	1	0	1	Overscan color (Red Component)
		01	0	1	0	1	Overscan color (Green Component)
		10	0	1	0	1	Overscan color (Blue Component)
N/A	0x01	00	0	1	0	1	Cursor Color 1 Red Component
		01	0	1	0	1	Cursor Color 1 Green Component
		10	0	1	0	1	Cursor Color 1 Blue Component
N/A	0x02	00	0	1	0	1	Cursor Color 2 Red Component
		01	0	1	0	1	Cursor Color 2 Green Component
		10	0	1	0	1	Cursor Color 2 Blue Component
N/A	0x03	00	0	1	0	1	Cursor Color 3 Red Component
		01	0	1	0	1	Cursor Color 3 Green Component
		10	0	1	0	1	Cursor Color 3 Blue Component
0	0x000-0x1FF	N/A	1	0	1	1	Cursor RAM Array, plane 0
1	0x200-0x3FF	N/A	1	0	1	1	Cursor RAM Array, plane 1

Table 2. Address Register Operation and AutoIncrementing.

## Frame Buffer Pixel Port Interface

There are 64 input pins P0-P7 (A-H) used to interface to the graphics and video frame buffer memories. The assignment of pins to input pixels is determined by the operation mode and multiplex rate.

## Video Port Clocking

Video data is synchronously clocked into Bt885 with the VLCLK input. VLCLK may be asynchronous from the pixel and/or graphics load clock, as an internal FIFO is used to synchronize video data to graphics pixel data.

Two status signals are available to control the loading of video pixel data into Bt885: VALID and READY. VALID is provided by the system to Bt885 and is asserted to indicate that valid video data is being presented on the video pixel port. The READY signal is an output from Bt885 that indicates that it is accepting pixel data. For data to be accepted on any particular VLCLK rising edge, both the VALID and READY signals must be high through the clock edge.

The system must load video data into Bt885 prior to the time that it is to be used. In systems where there is a one-to-one relationship between video pixels and graphics pixels in the frame buffer and this data is delivered simultaneously, the FIFO operation can be ignored and VALID would be tied to the pixel blanking signal from the graphics subsystem (BLANK\*). In this mode, the FIFO would never be filled and, therefore, READY may be ignored.

The internal video data FIFO is reset to an empty state on each detected vertical blank period. The system can immediately begin loading data into the video port regardless of the video window's position on the screen. If at any time the video FIFO is empty when video data is required, Status Register 2 bit SR27 will be set to one. The underflow bit will remain set until Status Register 2 is written, then SR27 will be cleared.

For proper operation of the video pipeline reset, VLCLK must be a free-running clock.

## Circuit Description (continued)

**VideoCache™ FIFO Operation**

The Bt885 provides a FIFO buffer for video pixels to allow for asynchronous video and graphics operation, and to ease system design requirements. Use of the VideoCache™ FIFO features is entirely optional and not necessary for synchronous designs.

**Loading VideoCache™ FIFO**

The VideoCache™ FIFO accepts a group of data (the exact number is given by the current video mode) when the following conditions are met on any single rising edge of VCLK:

1. The FIFO is ready to accept data (i.e., it is not full). This is determined by the state of the READY out signal.
2. The system is presenting data, indicating this to the CacheDAC™ by asserting the VALID signal with the data.

**Unloading VideoCache™ FIFO**

Bt885 will unload the VideoCache™ FIFO dependant on the setting of bit CR41. If CR41 = 1, the video will only be unloaded while Bt885 is scanning through the video window. If CR41 = 0, then video will always be unloaded during active graphics time. The unloading process is independent of color keying.

**General Purpose Signals****DIVCLK1 / DIVCLK2**

These signals provide programmable free-running clocks based on the internal pixel clock. They can be used to generate external pixel load clocks, such as VLCLK or GLCLK. A gated clock may be generated from DIVCLK1 by using another general purpose signal, SEN, described below.

**SEN**

This signal is used to provide a gating control for DIVCLK1. SEN can be programmed to start relative to the falling edge of internally detected vertical sync (see cursor operation) in units of scanlines and relative to the falling edge of HSYNC\* in DIVCLK1 cycles using the serial clock enable start (horizontal and vertical) registers. Duration is set in units of scanlines for the vertical direction and in DIVCLK1 cycles for the horizontal direction (relative to the beginning of SEN) using the serial clock enable duration (horizontal and vertical)

registers. This signal is guaranteed to transition only during DIVCLK1 low time.

This signal may be used, for example, to control a VRAM shift clock which runs during non-blanking time. When an appropriate delay is programmed from the leading edge of HSYNC\*, the serial data can be properly positioned before the trailing edge of BLANK\*. The SEN duration register then stops the serial clock to allow the system to perform VRAM row data transfer. Because \*HSYNC is sampled with the internal pixel clock, there may be an additional pixel clock delay between \*HSYNC falling and the SEN rising.

**HFULL**

This signal is asserted when the VideoCache™ FIFO is more than half full.

**Video Window Operation**

The XSTART register indicates the starting X position on the screen for the video window relative to the ENABLE pin (Figure 2). A value of zero indicates that the video window begins with the first (leftmost) pixel of each horizontal scan line. The YSTART register indicates the starting Y position on the screen for the video window. A value of zero indicates that the video window begins on the first scan line of each frame. The XWIDTH register indicates the number of pixels per scan line within the video window. A value of zero indicates that there are no pixels in the video window. The YHEIGHT register indicates the number of scan lines within the video window. A value of zero indicates that there are no scan lines in the video window.

All four values, XSTART, XWIDTH, YSTART, and YHEIGHT should be written sequentially. Internal video window coordinates are loaded during the next detected blanking interval after the YHEIGHT register is written.

**Video Scaling Operation**

Horizontal upscaling is accomplished by using the output of an overflowing 12-bit accumulator to either clock a value out of the VideoCache™ FIFO to the DACs or to hold the current DAC value. At the start of each scan line, the accumulator is initialized to the value stored in the XSCALEINT register.

On each pixel, the value stored in the XSCALEINC register is added to the accumulator.

If the addition results in a carry, a pixel is clocked out of the VideoCache™ FIFO to the DACs. If no carry

## Circuit Description (continued)

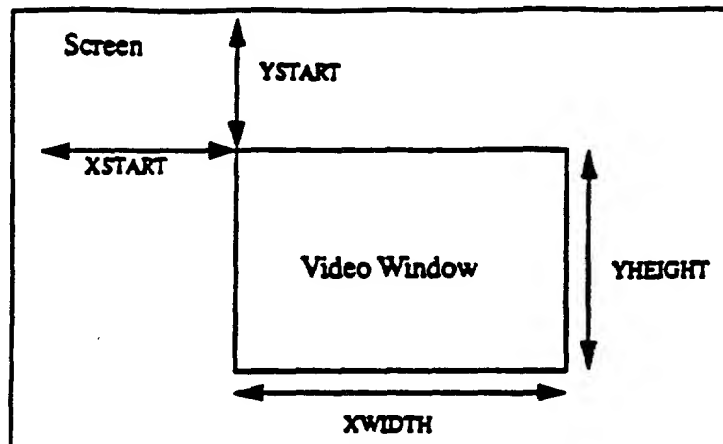


Figure 2. Video Window Registers.

occurs, the previous DAC value is held. This style of scaling is known as a Digital Differential Algorithm (DDA).

To accomplish scaling, the system supplying Bt885 with video pixels must precalculate the DDA constants required for the desired scale factor and load the values into the two 12-bit X-scaling registers, XSCALEINTT and XSCALEINC, as follows:

$$\text{XSCALEINC} = [(\text{Source Video Width} * 0x1000) + 0x0800 - \text{XSCALEINTT}] / \text{Destination Video Width}$$

$0 \leq \text{XSCALEINTT} \leq 0x0fff$ . XSCALEINTT can be used to set the replication phase of the DDA in more advanced applications. Commonly, XSCALEINTT is set to 0x0800.

### Color Key Operation

Selection between the video and graphics pixel data may be based on a specified range of graphic pixel values. A "color key set" may be defined which specifies one or more graphic pixel values that allow video pixels to be shown.

To define the color key set, three color key registers and three color mask registers are used. A graphic pixel value is bitwise XORed with the color key and the result is NANDed with the color mask. If the result is

one, the corresponding video pixel is displayed in its place.

When a graphic pixel value falls within the color key set, the corresponding video pixel is displayed rather than the graphic pixel. Color key detection may occur either before the palette lookup or after the palette lookup. In 16- and 24-bit pixel modes, if palette bypass is enabled, selecting matching after the palette matches based on the actual values that would be applied to the DACs.

When matching after the palette, bit CR42 of Command Register 4 should be set to 1, and the color key registers and color mask registers represent 24-bit RGB values each. The registers are ordered with red at the lowest address, then green and blue.

When matching before the palette, bit CR42 of Command Register 4 should be set to zero. The color key registers and color mask registers represent unmultiplied graphic pixel values, with the red register as the least significant byte, then green and blue. Only the bits needed to represent the pixel are used. For example, an 8-bit pixel color key and mask use only the red registers, 16-bit pixels use only the red and green registers.

Pixel selection occurs only within the current video window boundaries, and only when bit CR46 of Command Register 4 is set to 0 to allow color key detection. When CR46 is set to 1, all pixels within the video win-

## Circuit Description (continued)

dow will display the video pixels, regardless of color mask and key register values.

The hardware cursor always has display priority over color key selection.

**Example 1**

Match a specific 8-bit pseudo-color palette position (value 0xFE).

CR42 = 0 (matching before palette)

CR46 = 0 (allow color keying)

Color Mask: (B) 0xFF (G) 0xFF (R) 0xFF

Color Key: (B) 0xFF (G) 0xFF (R) 0xFE

**Example 2**

Match a range of blue values between 0xC0 and 0xC7.

CR42 = 1 (matching after palette)

CR46 = 0 (allow color keying)

Color Mask: (B) 0xF8 (G) 0x00 (R) 0x00

Color Key: (B) 0xC0 (G) 0x00 (R) 0x00

**Example 3**

Use bit 15 in a TARGA 15-bit true color mode to perform color key.

CR42 = 1 (matching after palette)

CR46 = 0 (allow color keying)

Color Mask: (B) 0x00 (G) 0x80 (R) 0x00

Color Key: (B) 0x00 (G) 0x80 (R) 0x00

**YCrCb-to-RGB Matrix**

The matrix converts the YCrCb video data to 24 bits of RGB data (8 bits each).

The YCrCb-to-RGB conversion is compliant with CCIR Recommendation 601-1 as follows:

$$R = 1.164(Y - 16) + 1.596(Cr - 128)$$

$$G = 1.164(Y - 16) - 0.813(Cr - 128) - 0.391(Cb - 128)$$

$$B = 1.164(Y - 16) + 2.018(Cb - 128)$$

Need to rev. this to be 100%

## Circuit Description (continued)

**Modes of Operation—Graphics****4-Bits/Pixel Operation (8:1 MUX)**

The 32 input bits are multiplexed 8:1 and configured for 4 bits/pixel. There are eight independent 4-bit pixel ports, P7:4 (A-D) and P3:0 (A-D). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every eight pixel clock cycles. The 4 bits from each port will select one of sixteen locations in the palette (see Table 10 in the Internal Registers Section).

**8-Bits/Pixel Operation (4:1 MUX)**

The 32 input bits are multiplexed 4:1 and configured for 8 bits/pixel. There are four independent 8-bit pixel ports, (A-D). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every four pixel clock cycles. The 8 bits from each port will select 1 of 256 locations in the palette (see Table 10 in the Internal Registers Section).

**8-Bits/Pixel Operation (2:1 MUX)**

The 16 input bits are multiplexed 2:1 and configured for 8 bits/pixel. There are two independent 8-bit pixel ports, (A-B). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every two pixel clock cycles. The 8 bits from each port will select 1 of 256 locations in the palette (see Table 10 in the Internal Registers Section).

**8-Bits/Pixel Operation (1:1 MUX)**

The 8 input bits are multiplexed 1:1 and configured for 8 bits/pixel. There is one 8-bit pixel port, (A). The pixel bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every pixel clock cycle. The 8 bits will select 1 of 256 locations in the palette (see Table 10 in the Internal Registers Section).

**16-Bits/Pixel Operation (2:1 MUX)**

The 32 input bits are multiplexed 2:1 and configured for 16 bits/pixel. There are two independent 16-bit pixel ports, (A-B) and (C-D). The bits are latched on the rising edge of GLCLK. One rising edge of GLCLK should occur every two pixel clock cycles. The pixel bits multiplexed in this mode are from the same ports of RGB color formats of 5:5:5 or 5:6:5. P7D and P7B are ignored internally when the 5:5:5 color format is selected (see Table 10 in the Internal Registers Section).

Bit CR24 in Command Register 2 can be programmed to enable or disable true-color palette bypass. When the

bypass mode is selected, the pixel data bypasses the palette as well as the pixel mask, and is transferred to the proper MSBs of the respective DACs, the remaining LSBs are set to zeros. When the bypass mode is not selected, the pixel data indexes the palette, and color information is passed to the respective DACs. Bit CR22 in Command Register 2 determines whether palette addressing is sparse or contiguous. For sparse palette addressing, each independent color component of pixel data is mapped to the most significant bits of the respective palette address; the least significant bits are set to zero. For contiguous palette addressing, each independent color component of the pixel data is mapped to the least significant bits of the respective palette address; the most significant bits are set to zero. The color palette values indexed, for either sparse or contiguous addressing, are transferred to the DACs. When 5:5:5 or 5:6:5 color format is selected, the display can contain 32 K or 64 K simultaneous color. The DACs can be configured for 6 or 8 bits of resolution in this mode. If 5:5:5 color format is selected, the most significant bit may be used for color key operation (see Tables 3 and 4).

**16-Bits/Pixel Operation (1:1 MUX)**

The 16-bit pixel port (A-B) is latched on the rising edge of GLCLK and is multiplexed 1:1. One rising edge of GLCLK should occur every pixel clock cycle.

Bit CR25 in Command Register 2 can be programmed to enable or disable true-color palette bypass. When the bypass mode is selected, the pixel data bypasses the palette as well as the pixel mask, and is transferred to the proper MSBs of the respective DACs. When the bypass mode is not selected, the pixel data indexes the proper locations in the palette, and the correct color information is passed to the respective DACs. Bit CR22 in Command Register 2 determines whether palette addressing is sparse or contiguous. For sparse palette addressing, each independent color component of pixel data is mapped to the most significant bits of the respective palette address; the least significant bits are set to zero. For contiguous palette addressing, each independent color component of the pixel data is mapped to the least significant bits of the respective palette address; the most significant bits are set to zero. The color palette values indexed, for either sparse or contiguous addressing, are transferred to the DACs. When 5:5:5 or 5:6:5 color format is selected, the display can contain 32K or 64K simultaneous colors. The DACs can be configured for 6 or 8 bits of resolution in this mode (see Table 5).

If 5:5:5 color format is selected, the most significant bit is may be used for color key operation.



## Circuit Description (continued)

## 24-Bits/Pixel Operation (1:1 MUX)

When 24 bits per pixel in 1:1 MUX mode is selected, there is one 24-bit pixel port, (A-C). The pixel bits are latched on the rising edge of GLCLK and multiplexed 1:1. One rising edge of GLCLK should occur every pixel clock cycle. The RGB color format in this mode is 8:8:8.

Bit CR25 in Command Register 2 can be programmed to enable or disable true-color palette bypass. When the bypass mode is selected, the pixel data bypasses the palette as well as the pixel mask, and is transferred to the proper MSBs of the respective DACs. When the bypass mode is not selected, the pixel data indexes the proper locations in the palette, and the independent RGB color values are passed to the respective DACs (see Table 7a-7c). When 8:8:8 color format is selected, the display can contain 16.8 million simultaneous colors. The DACs should be configured for 8 bits of resolution in this mode (CR25 = 1, CR01 = 1). CR41 and CR40 can be used to alter the pixel read order to BRG or BGR.

## Pixel Read Mask Register

The pixel data can be masked before being transferred to the color palette with the 8-bit pixel mask register. The pixel data is bit-wise logically ANDed with the contents of the pixel read mask register. The result is used to address the color palette RAM. The addressed location provides 24 bits of color information to the three D/A converters. Pixel masking is enabled for all modes of operation except when the true-color bypass is enabled. The pixel mask register is initialized to logical ones at reset (see the Table 13, Register Values on Reset Table in the Internal Register section).

## Modes of Operation—Video

## Big-Endian versus Little-Endian Pixel Display Order

Pixel display order may be either big endian or little endian. The display order is selected by setting bit CR43 in Command Register 4 to a zero for little endian, or a one for big endian. The pixel ordering and YCrCb-to-RGB conversions are shown in Figures 3-5 and the video pixel port configuration is shown in Table 9. The following descriptions of the modes of operation of the video pixel port are based on little endian display order.

## YCrCb 1:0.5:0.5 Operation (4 Byte/8 Pixels)

The 32 input bits are configured for YCrCb 1:0.5:0.5. There are four independent 8-bit pixel ports, (E-H). Each group of four bytes results in eight output pixels. The pixel bits are latched on the rising edge of VLCLK.

## YCrCb 1:0.5:0.5 Operation (2 Byte/4 Pixels)

The 16 input bits are configured for YCrCb 1:0.5:0.5. There are two independent 8-bit pixel ports, (G-H). Each group of two bytes results in four output pixels. The pixel bits are latched on the rising edge of VLCLK.

## YCrCb 1:0.5:0.5 Operation (1 Byte/2 Pixels)

The 8 input bits are configured for YCrCb 1:0.5:0.5. There is one 8-bit pixel port, (H). Each byte loaded results in two output pixels. The pixel bits are latched on the rising edge of VLCLK.

## YCrCb 2:1:1 Operation (4 Byte/4 Pixels)

The 32 input bits are configured for YCrCb 2:1:1. There are four independent 8-bit pixel ports, (E-H). The pixel bits are latched on the rising edge of VLCLK.

## YCrCb 2:1:1 Operation (2 Byte/2 Pixels)

The 16 input bits are configured for YCrCb 2:1:1. There are two independent 8-bit pixel ports, (G-H). The pixel bits are latched on the rising edge of VLCLK.

## YCrCb 2:1:1 Operation (1 Byte/1 Pixel)

The 8 input bits are configured for YCrCb 2:1:1. There is one 8-bit pixel port, (H). The pixel bits are latched on the rising edge of VLCLK.

## YCrCb 4:2:2 Operation (4 Byte/2 Pixels)

The 32 input bits are configured for 4:2:2. There are two independent 16-bit pixel ports, (E-F) and (G-H). The bits are latched on the rising edge of VLCLK.

## YCrCb 4:2:2 Operation (2 Byte/1 Pixel)

The 16 input bits are configured for YCrCb 4:2:2. There is one 16-bit pixel port, (G-H). The input bits are latched on the rising edge of VLCLK.

## Circuit Description (continued)

Bit	MSB															LSB
Format	X	R	R	R	R	R	G	G	G	G	G	B	B	B	B	B
Port 1	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A
Port 2	P7D	P6D	P5D	P4D	P3D	P2D	P1D	P0D	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C

Note: X bit may be used for color key before the palette.

Table 3. 5:5:5 RGB Graphics Color Format for Both 2:1 and 1:1 Multiplexing Modes.

Bit	MSB															LSB
Format	R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B
Port 1	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A
Port 2	P7D	P6D	P5D	P4D	P3D	P2D	P1D	P0D	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C

Table 4. 5:6:5 RGB Graphics Color Format for Both 2:1 and 1:1 Multiplexing Modes.

	MSB								LSB	XsMap to Zero
Pixel Mask Register	7	6	5	4	3	2	1	0		Register Bits
4 Bits/Pixel	x	x	x	x	3	2	1	0		Palette Index
8 Bits/Pixel	7	6	5	4	3	2	1	0		Palette Index
16 Bits/Pixel 5:5:5 Format SPARSE	7 7 7	6 6 6	5 5 5	4 4 4	3 3 3	x x x	x x x	x x x		Red Palette Index Green Palette Index Blue Palette Index
16 Bits/Pixel 5:5:5 Format CONTIGUOUS	x x x	x x x	x x x	4 4 4	3 3 3	2 2 2	1 1 1	0 0 0		Red Palette Index Green Palette Index Blue Palette Index
16 Bits/Pixel 5:6:5 Format SPARSE	7 7 7	6 6 6	5 5 5	4 4 4	3 3 3	x 2 x	x x x	x x x		Red Palette Index Green Palette Index Blue Palette Index
16 Bits/Pixel 5:6:5 Format CONTIGUOUS	x x x	x x x	x 5 x	4 4 4	3 3 3	2 2 2	1 1 1	0 0 0		Red Palette Index Green Palette Index Blue Palette Index
24 Bits/Pixel 8:8:8 Format	7 7 7	6 6 6	5 5 5	4 4 4	3 3 3	2 2 2	1 1 1	0 0 0		Red Palette Index Green Palette Index Blue Palette Index

(Note: x means final DAC bit will be 0)

Table 5. Graphics Pixel Index Masking.

## Circuit Description (continued)

Bit	MSB																								LSB
Format	R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G	B	B	B	B	B	B	B	B	B
Port 1	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A	

Table 6a. 24-bits/Pixel Graphics RGB Color Format (CR40,41 = 00) for 1:1 MUX Modes.

Bit	MSB																								LSB
Format	B	B	B	B	B	B	B	B	R	R	R	R	R	R	R	R	G	G	G	G	G	G	G	G	G
Port 1	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A	

Table 6b. 24-bits/Pixel Graphics BRG Color Format (CR41,40 = 01).

Bit	MSB																								LSB
Format	B	B	B	B	B	B	B	B	G	G	G	G	G	G	G	G	R	R	R	R	R	R	R	R	R
Port 1	P7C	P6C	P5C	P4C	P3C	P2C	P1C	P0C	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A	

Table 6c. 24-bits/Pixel Graphics BGR Color Format (CR41,40 = 10).

**16-Bits/Pixel 5:5:5 Operation (2:1 MUX)**

The 32 input bits are configured for 16 bits/pixel. There are two independent 16-bit pixel ports, (E-F) and (G-H). The bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:5:5. The most significant bit is not used.

**16-Bits/Pixel 5:5:5 Operation (1:1 MUX)**

The 16 input bits are configured for 16 bits/pixel. There is one 16-bit pixel port, (G-H). The input bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:5:5. The most significant bit is not used.

**16-Bits/Pixel 5:6:5 Operation (2:1 MUX)**

The 32 input bits are configured for 16 bits/pixel. There are two independent 16-bit pixel ports, (E-F) and (G-H). The bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:6:5.

**16-Bits/Pixel 5:6:5 Operation (1:1 MUX)**

The 16 input bits are configured for 16 bits/pixel. There is one 16-bit pixel port, (G-H). The input bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 5:6:5.

**24-Bits/Pixel Operation (1:1 MUX)**

The 24 input bits are configured for 24 bits/pixel. There is one 24-bit pixel port, (F-H). The bits are latched on the rising edge of VLCLK. The RGB color format in this mode is 8:8:8.

## Circuit Description (continued)

## Video Data Stream Length

Dependant on the video pixel interpolation selected in Bt885, a certain number of input bytes will represent a fixed number of output pixels. In general, Bt885 is capable of stretching an arbitrary-length horizontal input stream of *A* bytes into an arbitrary horizontal window of *B* RGB pixels.

The system may take advantage of the VideoCache™ FIFO to lower video bandwidth and frame buffer requirements. This is achieved by loading only the number of bytes of video data required for the displayed video window.

There are two parameters that can be adjusted in Bt885 to achieve this. One is the current video mode and the other is the scale factor that is applied to the video data.

To calculate the X scaling increment, use the following equations:

$$\begin{aligned} \text{Destination Video Width} = \\ & \left[ \frac{(\text{Source Video Width} \cdot 4096)}{(\text{XSCALEINC} + 1)} \right] \end{aligned}$$

i.e.,

$$\text{XSCALEINC} = \left[ \frac{(\text{Source Video Width} \cdot 4096)}{\text{Destination Video Width}} \right] - 1$$

If scaling is not required, both XSCALEINT and XSCALEINTT should be set to 4095.

Video Mode	Number of Input Bytes Required for X Source Pixels Independent of Scale
4:2:2	$(\text{INT}[(X + 1) / 2] + 1) \cdot 4$
2:1:1	$(\text{INT}[(X + 3) / 4] + 1) \cdot 4$
1:0:5:0:5	$(\text{INT}[(X + 7) / 8] + 1) \cdot 4$
15/16-bit RGB	$(\text{INT}[(X + 1) / 2] + 1) \cdot 4$
24-bit RGB	$(\text{INT}[(3 \cdot X + 3) / 4] + 1) \cdot 4$

The number of output pixels will depend on the values of XSCALEINTT and XSCALEINC.

The Bt885 always performs linear interpolation of YCrCb values. This requires that enough data must be delivered to perform the interpolation of the final pixel of a scan line. For example, generating four YCrCb 4:2:2 pixels would require the following 12-byte data stream:

[Cr0 Y0 Cb0 Y1] [Cb2 Y2 Cr2 Y3] [Cb4 Y4 Cr4 Y5]

In this case, the Cb4 and Cr4 values are used to perform the interpolation of the chroma value of the fourth pixel.

Video pixels are always removed from the VideoCache™ FIFO in 4-byte blocks. At the end of each video line, the video interpolator will be purged. Therefore, any remaining data in the interpolator at the end of a video output line up to the next 4-byte boundary will be ignored. In the example above, the Y4 would not be used and the Y5 value would be skipped.

Example: Scaling a 320 pixel wide CCIR601 4:2:2 format image to 640 pixels horizontally:

$$\begin{aligned} \text{Number of input bytes} = \\ \text{INT}[(320 + 1) / 2] + 1 \cdot 4 = 644 \end{aligned}$$

$$\begin{aligned} \text{XSCALEINC} = & \left[ \frac{(320 \cdot 4096)}{640} \right] - 1 = \\ & 2047 = 0 \times 07FF \end{aligned}$$

$$\text{XSCALEINTT} = \text{XSCALEINC} / 2 = 0 \times 03FF$$

## DAC Values in 16-Bits/Pixel Video Modes

In order to achieve 8-bit full-scale DAC output in the 5:5:5 16-bits/pixel video modes, each 5-bit value will be used as the five most significant bits of the 8-bit DAC value and the three most significant bits of the 5-bit pixel value will be duplicated in the low order 3 bits before the pixel value is passed to the DACs. Similarly, in 5:6:5 modes, when processing the 6-bit green component, the 6-bit value will be used as the 6 most significant bits of the 8-bit DAC value and the two most significant bits of the 6-bit pixel value will be duplicated in the low order 2 bits before the pixel value is passed to the DACs.

## Circuit Description (continued)

Bk	MSB															LSB
Format	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

YCrCb 1:0.5:0.5 Video Color Format (4 Bytes / 8 Pixels)

Bk	MSB															LSB
Format	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

YCrCb 1:0.5:0.5 Video Color Format (2 Bytes / 4 Pixels).

Bk	MSB															LSB
Format	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	X	X	X	X	X	X	X	X
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

YCrCb 1:0.5:0.5 Video Color Format (1 Byte / 2 Pixels).

Bk	MSB															LSB
Format	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

YCrCb 2:1:1 Video Color Format (4 Bytes / 4 Pixels)

## Circuit Description (continued)

Bit	MSB																LSB
Format	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	

YCrCb 2:1:1 Video Color Format (2 Bytes / 2 Pixels).

Bit	MSB																LSB
Format	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	Cb/Y/ Cr	X	X	X	X	X	X	X	X
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	

YCrCb 2:1:1 Video Color Format (1 Byte / 1 Pixel).

Bit	MSB																LSB
Format	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Cb	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	
Format	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	

YCrCb 4:2:2 Video Color Format (4 Bytes / 2 Pixels).

Bit	MSB																LSB
Format	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Cb/Cr	Y	Y	Y	Y	Y	Y	Y	Y
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G	
Format	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E	

YCrCb 4:2:2 Video Color Format (2 Bytes / 1 Pixel).

## Circuit Description (continued)

BH	MSB															LSB
Format	X	B	B	B	B	B	G	G	G	G	G	R	R	R	R	R
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

**5:5:5 BGR Video Color Format for Both 2:1 and 1:1 Multiplexing Modes.**

BH	MSB															LSB
Format	B	B	B	B	B	G	G	G	G	G	G	R	R	R	R	R
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

**5:5:5 BGR Video Color Format for Both 2:1 and 1:1 Multiplexing Modes.**

BH	MSB															LSB
Format	B	B	B	B	B	B	B	B	G	G	G	G	G	G	G	G
Port 1	P7H	P6H	P5H	P4H	P3H	P2H	P1H	P0H	P7G	P6G	P5G	P4G	P3G	P2G	P1G	P0G
Format	R	R	R	R	R	R	R	R	X	X	X	X	X	X	X	X
Port 2	P7F	P6F	P5F	P4F	P3F	P2F	P1F	P0F	P7E	P6E	P5E	P4E	P3E	P2E	P1E	P0E

**24-Bit BGR Video Color Format for 1:1 Multiplexing Modes.**

## Circuit Description (continued)

CCIR601 1:0.5:0.5

CCIR656 Component Ordering

Color Space: YCrCb

Subsampling: 1:0.5:0.5

Address (8N+)	0	1	2	3	4	5	6	7
Value	Cb0	Y0	Cr0	Y4	Cb8	Y8	Cr8	Y12

Pixel	0	1	2	3	4	5	6	7
Y	Y0	$\frac{3Y0 + Y4}{4}$	$\frac{Y0 + Y4}{2}$	$\frac{Y0 + 3Y4}{4}$	Y4	$\frac{3Y4 + Y8}{4}$	$\frac{Y4 + Y8}{2}$	$\frac{Y4 + 3Y8}{4}$
Cr	Cr0	Cr0	$\frac{3Cr0 + Cr8}{4}$	$\frac{3Cr0 + Cr8}{4}$	$\frac{Cr0 + Cr8}{2}$	$\frac{Cr0 + Cr8}{2}$	$\frac{Cr0 + 3Cr8}{4}$	$\frac{Cr0 + 3Cr8}{4}$
Cb	Cb0	Cb0	$\frac{3Cb0 + Cb8}{4}$	$\frac{3Cb0 + Cb8}{4}$	$\frac{Cb0 + Cb8}{2}$	$\frac{Cb0 + Cb8}{2}$	$\frac{Cb0 + 3Cb8}{4}$	$\frac{Cb0 + 3Cb8}{4}$

Figure 3. CCIR601 1:0.5:0.5 Video Format.

CCIR601 2:1:1

CCIR656 Component Ordering

Color Space: YCrCb

Address (8N+)	0	1	2	3	4	5	6	7
Value	Cb0	Y0	Cr0	Y2	Cb4	Y4	Cr4	Y6

Pixel	0	1	2	3	4	5	6	7
Y	Y0	$\frac{Y0 + Y2}{2}$	Y2	$\frac{Y2 + Y4}{2}$	Y4	$\frac{Y4 + Y6}{2}$	Y6	$\frac{Y6 + Y8}{2}$
Cr	Cr0	$\frac{3Cr0 + Cr4}{4}$	$\frac{Cr0 + Cr4}{2}$	$\frac{Cr0 + 3Cr4}{4}$	Cr4	$\frac{3Cr4 + Cr8}{4}$	$\frac{Cr4 + Cr8}{2}$	$\frac{Cr4 + 3Cr8}{4}$
Cb	Cb0	$\frac{3Cb0 + Cb4}{4}$	$\frac{Cb0 + Cb4}{2}$	$\frac{Cb0 + 3Cb4}{4}$	Cb4	$\frac{3Cb4 + Cb8}{4}$	$\frac{Cb4 + Cb8}{2}$	$\frac{Cb4 + 3Cb8}{4}$

Figure 4. CCIR601 2:1:1 Video Format.



Circuit Description (continued)

CCIR601 4:2:2  
CCIR656 Component Ordering  
Color Space: YCrCb

Address (8N+)	0	1	2	3	4	5	6	7
Value	Cb0	Y0	Cr0	Y1	Cb2	Y2	Cr2	Y3

Pixel	0	1	2	3
Y	Y0	Y1	Y2	Y3
Cr	Cr0	$\frac{Cr0 - Cr2}{2}$	Cr2	$\frac{Cr2 + Cr4}{2}$
Cb	Cb0	$\frac{Cb0 + Cb2}{2}$	Cb2	$\frac{Cb2 + Cb4}{2}$

Figure 5: CCIR601 4:2:2 Video Format.

## Circuit Description (continued)

## Cursor Operation

The Bt885 has an on-chip, three-color, 64 x 64 x 2 pixel user-definable cursor. This cursor works with both interlaced and noninterlaced systems. The cursor always has display priority over both video and graphics pixels.

The pattern for the cursor is provided by the cursor RAM, which may be accessed by the MPU at any time. The cursor is positioned through the cursor position register (Xp,Yp) (see Figure 6). A (0,0) written to the cursor position registers will place the cursor completely offscreen. A (1,1) written to the cursor position registers will place the lower right pixel of the cursor on the upper left corner of the screen. Only one cursor pattern per frame is displayed at the location specified for both interlaced and noninterlaced display formats, regardless of the number of updates to (Xp,Yp). The cursor's vertical or horizontal location is not affected during any frame displayed.

There are no restrictions on updating (Xp, Yp) other than both cursor position registers must be written when the cursor location is updated. Internal x and y position registers are loaded after the upper byte of Yp has been written to ensure one cursor pattern per frame at the correct location. The cursor pattern is displayed at the last cursor location written. Cursor positioning is relative to ENABLE. The cursor position is not

dependent upon BLANK\* (see Figure 6). The cursor Xp position is relative to the first rising edge of GLCLK when ENABLE is sampled at logical one. The cursor Yp position is relative to the first rising edge of GLCLK when ENABLE is sampled at logical one after the ENABLE vertical blanking interval has been determined (see Figure 6). If an ENABLE transition from logical zero to logical one (as determined by GLCLK) does not occur within 2048 internal pixel clocks, ENABLE is in vertical blanking.

For proper cursor operation, selection of interlaced or non-interlaced cursor display must be set using bit CR23 in Command Register 2.

Figure 7 is a visual explanation of planar pixel format and cursor RAM array pixel mapping.

While the cursor may be disabled by setting bits CR20-21 of Command Register 2 to zero, this practice is not recommended. The recommended method for disabling the cursor is to move it entirely offscreen by setting the cursor X and Y location registers to (0,0).

## Cursor Color Support

The cursor has three modes for color selection. Bits CR21 and CR20 in Command Register 2 determine which cursor mode is to be used. Mode 1 is a three-color cursor, Mode 2 is a Microsoft Windows™ cursor, and Mode 3 is an X-Windows cursor (see Table 7).

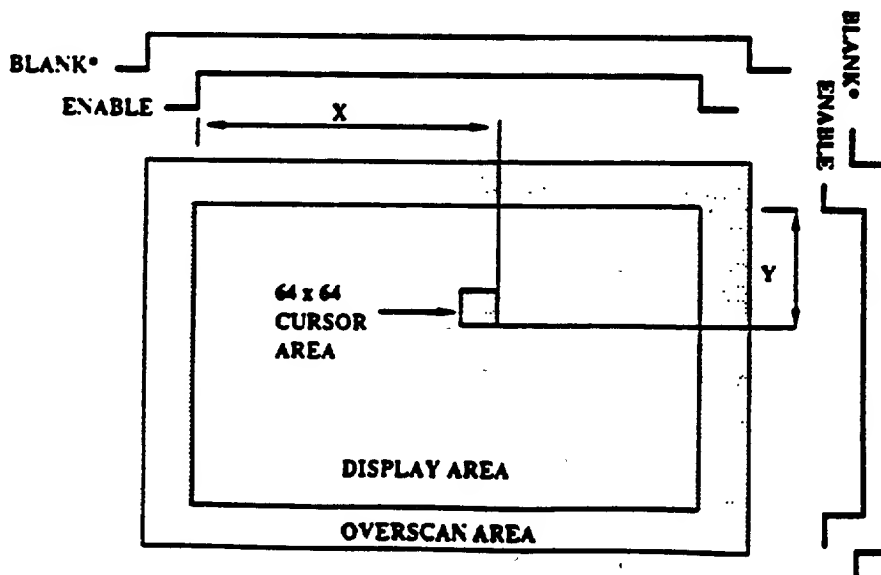


Figure 6. Cursor Positioning.

## Circuit Description (continued)

### Highlight Logic

The highlight logic is enabled in cursor mode 2 when plane 1 and plane 0 data are logical ones (see Table 7). When the highlight logic is enabled, it ensures that the graphics pixel highlighted has a unique color. This is because the highlight logic bit-wise complements the 24 (18)-bit graphics palette or bypass data supplied to the DACs.

### Video Generation

The HSYNC\* and BLANK\* inputs are latched on the rising edge of GLCLK to maintain synchronization with the pixel data.

Pipelined HSYNC\* and VSYNC\* are output on the HSYNC\* OUT and VSYNC\* OUT pins.

The CR05 command bit specifies whether a 0 or 7.5 IRE blanking pedestal is to be used. Command bits CR04, CR03, and CR02 specify whether the RGB outputs contain sync information.

Tables 8 and 9 detail how the HSYNC\* and BLANK\* inputs modify the output levels.

### SENSE\* Output

SENSE\* is a logical zero if one or more of the IOR, IOG, or IOB outputs have exceeded the internal voltage reference level of the SENSE\* comparator circuit. This output determines the presence of a CRT monitor and, with diagnostic code, the difference between a loaded or an unloaded RGB line can be discerned. The reference is generated by a voltage divider from the external 1.235 V voltage reference on the VREF pin. For the proper operation of the SENSE circuit, the following levels should be applied to the comparator with the IOR, IOG, and IOB outputs:

DAC Low Voltage  $\leq 260$  mV (see note below)

DAC High Voltage  $\geq 410$  mV (see note below)

There is an additional  $\pm 10\%$  tolerance on the above levels when the internal voltage reference is used. Both HSYNC\* and VSYNC\* should be a logical zero for SENSE\* to be stable. The SENSE\* output can drive only one CMOS load.

*Note:* SENSE values are subject to change upon completion of characterization.

## Circuit Description. (continued)

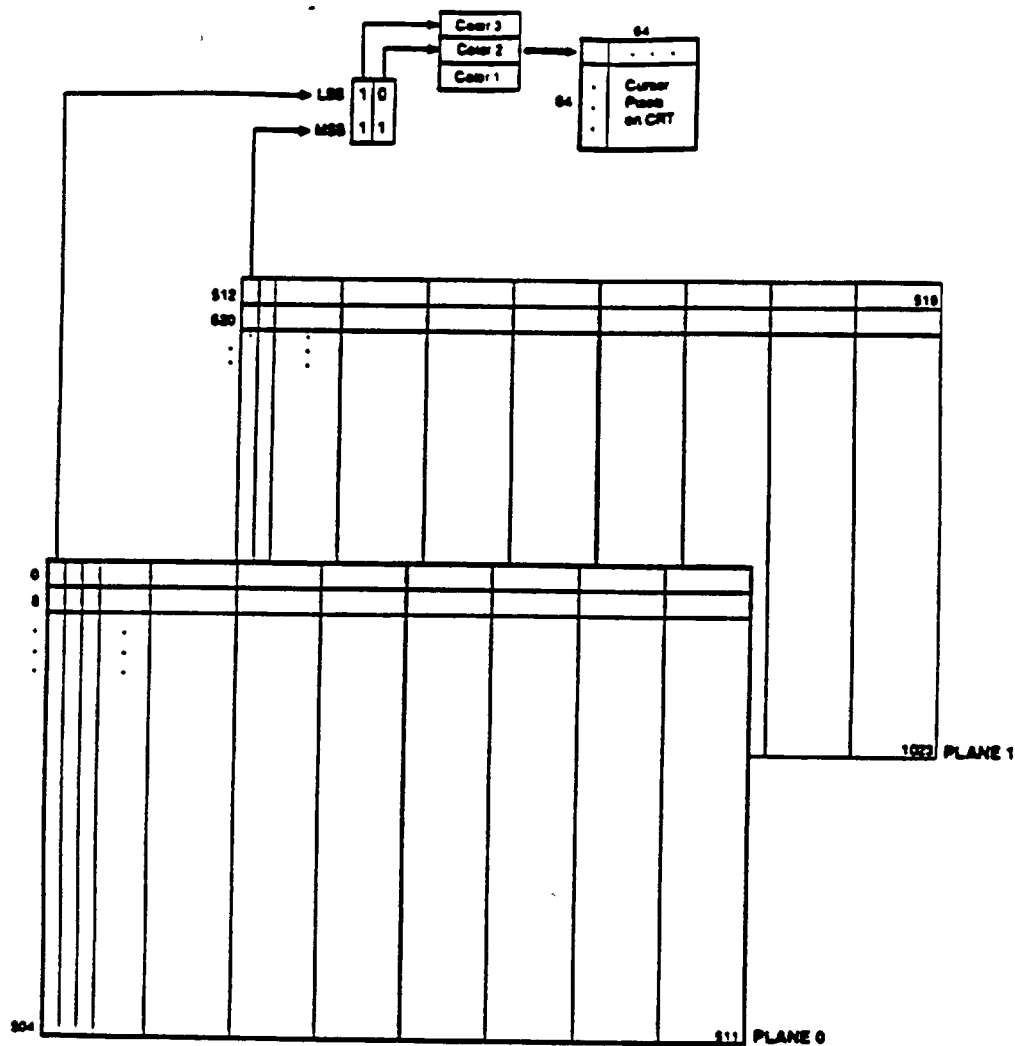
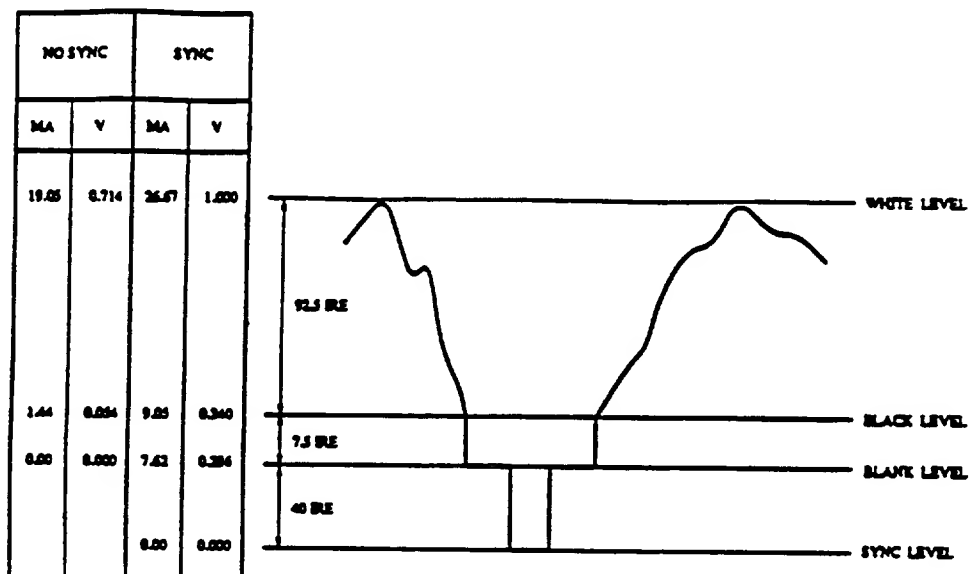


Figure 7. Planar Pixel Format and Cursor RAM Array Pixel Mapping.

Plane 1	Plane 0	MODE 1	MODE 2	MODE 3
0	0	Cursor not displayed	Cursor Color 1	Palette Data
0	1	Cursor Color 1	Cursor Color 2	Palette Data
1	0	Cursor Color 2	Palette Data	Cursor Color 1
1	1	Cursor Color 3	Highlight	Cursor Color 2

Table 7. Cursor Color Modes.

## Circuit Description (continued)



Note: 75  $\Omega$  doubly-terminated load, VREF = 1.235 V, and RSET = 147 $\Omega$ . RS-343A levels and tolerances assumed on all levels.

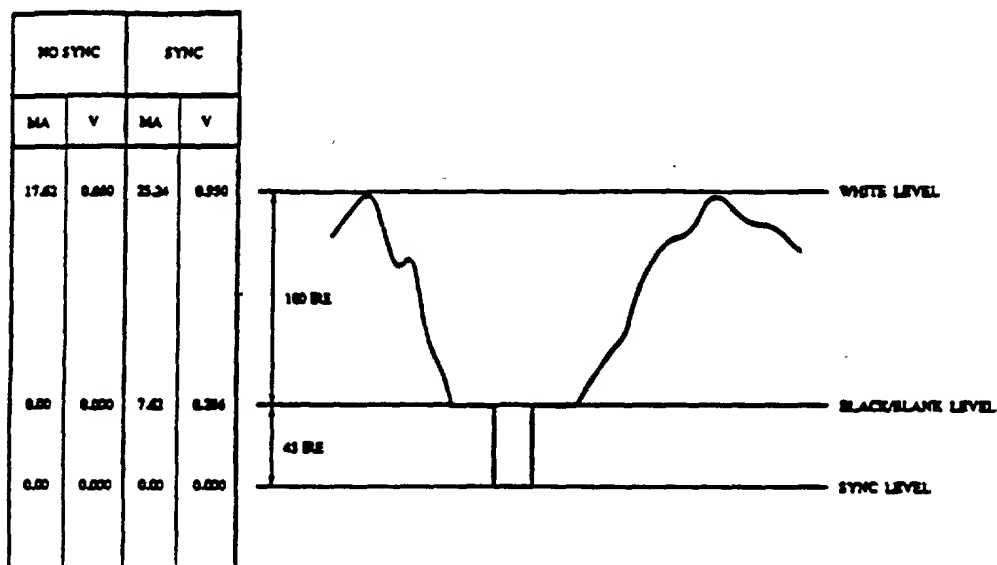
Figure 8. Composite Video Output Waveforms (SETUP = 7.5 IRE).

Description	Sync Disabled	Sync Enabled	HSYNC*	BLANK*	DAC Input Data
	Iout (mA)	Iout (mA)			
WHITE	19.05	26.67	1	1	0xFF
DATA	data + 1.44	data + 9.05	1	1	data
DATA-SYNC	data + 1.44	data + 1.44	0	1	data
BLACK	1.44	9.05	1	1	0x00
BLACK-SYNC	1.44	1.44	0	1	0x00
BLANK	0	7.62	1	0	xx
SYNC	0	0	0	0	xx

Note: 75  $\Omega$  doubly-terminated load, VREF = 1.235 V, and RSET = 147 $\Omega$ .

Table 8. Video Output Truth Table (SETUP = 7.5 IRE).

## Circuit Description (continued)



Note: 75  $\Omega$  doubly-terminated load, VREF = 1.235 V, and RSET = 147 $\Omega$ . RS-343A levels and tolerances assumed on all levels.

Figure 9. Composite Video Output Waveforms (SETUP = 0 IRE).

Description	Sync Disabled	Sync Enabled	HSYNC*	BLANK*	DAC Input Data
	Iout (mA)	Iout (mA)			
WHITE	17.62	25.24	1	1	0xFF
DATA	data	data + 7.62	1	1	data
DATA-SYNC	data	data	0	1	data
BLACK	0	7.62	1	1	0x00
BLACK-SYNC	0	0	0	1	0x00
BLANK	0	7.62	1	0	xx
SYNC	0	0	0	0	xx

Note: 75  $\Omega$  doubly-terminated load, VREF = 1.235 V, and RSET = 147 $\Omega$ .

Table 9. Video Output Truth Table (SETUP = 0 IRE).

## Internal Registers

## Command Register 0

This register may be written to or read by the MPU at any time and is not initialized at power-up. CR00 corresponds to data bus bit D0, the least significant data bit.

CR0 7	Reserved	This bit must be written with a 0 to ensure proper operation.
CR0 6	Clock Disable ANDed with CR00 (0) Normal Operation (1) Disable Internal Clocking	When this bit and CR00 are a logical one, the internal clock and output clocks are disabled to further conserve power when in power-down mode. The RAM still retains the data, and MPU reads and writes can occur with no loss of data. When this bit is a logical zero, internal clocking is enabled and output clocks will be generated.
CR0 5	Pedestal IRE (0) Disable (1) Enable 7.5 IRE	This bit determines the video blanking pedestal. A logical zero always sets a 0 IRE blanking pedestal and sets 7.5 IRE.
CR0 4 CR0 3 CR0 2	Blue Sync Enable Green Sync Enable Red Sync Enable (0) Disable Sync (1) Enable Sync	These bits specify whether the respective IOB, IOG, or IOR outputs are to contain sync information.
CR0 1	DAC 6/8-Bit Resolution (0) 6-bit Operation (1) 8-bit Operation	This bit specifies whether the MPU is reading and writing 8 bits (logical one) or 6 bits (logical zero) of color information each cycle.
CR0 0	Power-Down Enable (0) Normal Operation (1) Power-Down Operation	While this bit is a logical zero, the device operates normally. If this bit is a logical one, the DACs and power to the RAM and VideoCache™ FIFO are turned off. The RAM still retains the data, and CPU reads and writes can occur with no loss of data.  The DACs will be turned off during sleep mode only if a voltage reference (internal or external) is used.

## Internal Registers (continued)

## Command Register 1

This register may be written to or read by the MPU at any time. CR10 corresponds to data bus bit D0, the least significant data bit (see Table 10). All command register bits are set to logical zero upon asserting a low signal on the RESET<sup>™</sup> pin.

CR13	CR12	CR11	CR10	Pixel Latching Sequence	MUX Rate	Operating Modes
0	0	0	0	P7:0(A)	1:1	VGA 8-bits per pixel
0	0	0	1	P7:0(A) P7:0(B)	2:1	8-bits per pixel
0	0	1	0	P7:0(A) P7:0(B) P7:0(C) P7:0(D)	4:1	8-bits per pixel
0	0	1	1	P3:0(A) P7:4(A) P3:0(B) P7:4(B) P3:0(C) P7:4(C) P3:0(D) P7:4(D)	8:1	4-bits per pixel
1	0	0	1			Reserved
0	1	0	0	P7:0(B-A)	1:1	15-bits per pixel, 5:5:5
0	1	0	1	P7:0(B-A) P7:0(D-C)	2:1	15-bits per pixel, 5:5:5
0	1	1	0	P7:0(B-A)	1:1	16-bits per pixel, 5:6:5
0	1	1	1	P7:0(B-A) P7:0(B-C)	2:1	16-bits per pixel, 5:6:5
1	0	0	0	P7:0(C-A)	1:1	24-bits per pixel
1010-1111						Reserved

Table 10. Modes of Operation (Graphic Pixel Port Configuration).



## Internal Registers (continued)

CR17	CR16	CR15	CR14	Pixel Latching Sequence CR43 = 0	Pixel Latching Sequence CR43 = 1	Bytes Per VLCLK	Pixels Per VLCLK	Operating Modes
0	0	0	0	N/A	N/A	N/A	N/A	All Video Modes Disabled
0	0	0	1	P7:0(H)	P7:0(H)	1	2	CCIR601 YCrCb 1:0.5:0.5
0	0	1	0	P7:0(G) P7:0(H)	P7:0(H) P7:0(G)	2	4	CCIR601 YCrCb 1:0.5:0.5
0	0	1	1	P7:0(E) P7:0(F) P7:0(G) P7:0(H)	P7:0(H) P7:0(G) P7:0(F) P7:0(E)	4	8	CCIR601 YCrCb 1:0.5:0.5
0	1	0	0	P7:0(H)	P7:0(H)	1	1	CCIR601 YCrCb 2:1:1
0	1	0	1	P7:0(G) P7:0(H)	P7:0(H) P7:0(G)	2	2	CCIR601 YCrCb 2:1:1
0	1	1	0	P7:0(E) P7:0(F) P7:0(G) P7:0(H)	P7:0(H) P7:0(G) P7:0(F) P7:0(E)	4	4	CCIR601 YCrCb 2:1:1
0	1	1	1	P7:0(H-G)	P7:0(H-G)	2	1	CCIR601 YCrCb 4:2:2
1	0	0	0	P7:0(F-E) P7:0(H-G)	P7:0(H-G) P7:0(F-E)	4	2	CCIR601 YCrCb 4:2:2
1	0	0	1	P7:0(H-G)	P7:0(H-G)	2	1	15-bits per pixel, 5:5:5
1	0	1	0	P7:0(F-E) P7:0(H-G)	P7:0(H-G) P7:0(F-E)	4	2	15-bits per pixel, 5:5:5
1	0	1	1	P7:0(H-G)	P7:0(H-G)	2	1	15-bits per pixel, 5:6:5
1	1	0	0	P7:0(F-E) P7:0(H-G)	P7:0(H-G) P7:0(F-E)	4	2	15-bits per pixel, 5:6:5
1	1	0	1	P7:0(H-F)	P7:0(H-F)	3	1	24-bits per pixel
1110- 1111								Reserved

Table 11. Modes of Operation (Video Pixel Port Configuration).

**Internal Registers (continued)****Command Register 2**

This register may be written to or read by the MPU at any time. CR20 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET™ pin.

CR2 7	Reserved	This bit must be written with a 0 to ensure correct operation.
CR2 6	Reserved	This bit must be written with a 0 to ensure correct operation.
CR2 5	True-Color bypass Enable (0) Pixel Addresses Palette (1) Pixel Bypasses Palette	When this bit is a logical zero, the pixel palette is addressed by the pixel data. When this bit is a logical one, the RGB pixel data bypasses the color palette and drives the DACs directly. True-color bypassing is only available for pixel sizes of 16 and 24 bits.
CR2 4	Oscillator Select (0) OSC Selected (1) OSC <sup>+</sup> Selected	When this bit is a logical zero, OSC is selected as the TTL pixel clock input. When this bit is a logical one, OSC <sup>+</sup> is selected as the TTL pixel clock input.
CR2 3	Display Mode Select (0) Noninterlaced (1) Interlaced	When this bit is a logical zero, the display format is noninterlaced. When the bit is a logical one, the display format is interlaced. The mode must be set properly to ensure proper operation of the internal cursor.
CR2 2	16-Bit/Pixel Palette Index Select (0) Sparse Indexing (1) Contiguous Indexing	When this bit is a logical zero, palette addressing is sparse. The RGB color component pixel data is mapped to the most significant bits of the RGB palette address. The least significant of the palette address bits are set to (0). When this bit is a logical one, palette addressing is contiguous. The RGB color component pixel data is mapped to the least significant bits of the palette address. The most significant bits of the address are set to (0).
CR2 1,0	Cursor Mode Select (00) Cursor Disabled (01) 3-color cursor (10) 2-color/Microsoft Windows™ cursor (11) 2-color/X-Windows cursor	These bits determine the functionality of the onboard 64 x 64 x 2 hardware cursor.

## Internal Registers (continued)

## Accessing the Extended Registers

An extended register set is used to accommodate all features of the Bt885. Since there are only four register select lines (and all 16 combinations have already been used), the extended registers must be accessed indirectly.

For example, Command Register 3 is accessed with the following sequence of operations:

1. Set RS3–RS0 = 0000, Address Register.
2. Write Address Register to 0x02
3. Set RS3–RS0 = 1010 (Extended Address Register).
4. Read or Write Command Register 3.

Table 12 shows the indirect addressing mapping for each extended register.

Address Register Value	Extended Register Name
0x00	Status Register 1 (read only)
0x01	Status Register 2 (read/write)
0x02	Command Register 3
0x03	Command Register 4
0x04 – 0x05	Video Window XSTART-Low & High
0x06 – 0x07	Video Window YSTART-Low & High
0x08 – 0x09	Video Window XWIDTH-Low & High
0x0A – 0x0B	Video Window YHEIGHT-Low & High
0x0C – 0x0F	Reserved
0x10 – 0x11	XCALEINT-Low & High
0x12 – 0x13	XSCALEINC-Low & High
0x14 – 0x17	Reserved
0x18 – 0x19	Serial Clock Enable Start (Horizontal)-Low & High
0x1A – 0x1B	Serial Clock Enable Duration (Horizontal)-Low & High
0x1C – 0x1D	Serial Clock Enable Start (Vertical)-Low & High
0x1E – 0x1F	Serial Clock Enable Duration (Vertical)-Low & High
0x20	DIVCLK1 Rate
0x21	DIVCLK2 Rate
0x22	Reserved
0x23 – 0x25	Color Mask
0x26	Reserved
0x27 – 0x29	Color Key
0x2A – 0x2D	Reserved
0x2E	VideoCache™ FIFO Size
0x2F – 0xFF	Reserved

Table 12. Extended Registers Address Map. (RS3–RS0 = 1010)

## Internal Registers (continued)

## Command Register 3

This register may be written to or read by the MPU at any time. CR30 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET® pin.

CR3 7	MODE0 Input/Output Select (0) MODE0 Input (1) MODE0 Output	This bit determines if the MODE0 pin is configured as an input or an output.
CR3 6	Enable Internal Load Clock (0) Use GLCLK (1) Use Internal Load Clock	In applications where an external load clock is not provided, setting CR36 = 1 allows the internal load clock determined by the graphics mux rate to internally sample the graphics input pixels, blanking, horizontal, and vertical sync inputs. Setting CR36 = 0 causes Bt885 to sample these inputs on the basis of GLCLK pin.
CR3 5	DIVCLK2 Select (0) DIVCLK2 Enabled (1) DIVCLK2 Disabled	A logical zero must be written to this bit to enable the graphics divide-down clock, DIVCLK2, to be output. A logical one written to this bit three-states the DIVCLK2 output.
CR3 4	ECL Clock Select (0) TTL Level Clock Selected (1) Differential ECL Level Clock Selected	A logical one written to this bit enables the differential ECL clock input buffer using OSC and OSC* as inputs. A logical zero written to this bit disables the ECL clock buffer and allows OSC, GLCLK, or the 2x clock multiplier to directly drive the logic. If a logical one is written to this bit, then the clock multiplier and TTL clock selections are overridden. If CR34 = 1, then bit CR33 will be ignored.
CR3 3	2x Clock Multiplier Select (0) 2x Clock Multiplier Disabled. (1) 2x Clock Multiplier Enabled.	This bit enables or disables the 2x clock multiplier. A logical one written to this bit enables the onboard 2x TTL clock multiplier for high-speed operations. A logical zero written to this bit will disable the clock multiplier and will allow the external clock source to directly drive the logic. If CR34 = 1, then this bit will be ignored.
CR3 2	DIVCLK1 Select (0) DIVCLK1 Enabled (1) DIVCLK1 Disabled	A logical zero must be written to this bit to enable the video divide-down clock, DIVCLK1, to be output. A logical one written to this bit three-states the DIVCLK1 output. If DIVCLK1 Select is set to one, then the SEN output pin is three-stated as well.
CR3 1,0	MSBs for 10-bit Address Counter CR31 = A9 CR30 = A8	CR31 and CR30 are 2 MSBs of the 10-bit cursor address counter. To set this counter to access a particular location in the 64 x 64 x 2 cursor RAM array, these 2 bits must be written to Command Register 3 before the lower 8 bits are written to the address counter through the MPU port. As the 10-bit address counter autoincrements, the new values of this register can be read back through CR31 and CR30. The contents of this counter will be reset with the assertion of the external RESET pin.

## Internal Registers (continued)

## Command Register 4

This register may be written to or read by the MPU at any time and is not initialized at power-up. CR40 corresponds to data bus bit D0, the least significant data bit. All command register bits are set to logical zero upon asserting a low signal on the RESET<sup>™</sup> pin.

CR4	7	VideoCache <sup>™</sup> FIFO Reset (0) Normal Operation (1) Reset VideoCache <sup>™</sup> FIFO	A logical zero written to this bit, enables normal VideoCache <sup>™</sup> FIFO operation. A logical one written to this bit resets the VideoCache <sup>™</sup> FIFO after four video load clocks. Reset value is zero (normal video FIFO operation).
CR4	6	Color Key Override (0) Normal Color Key Operation (1) Video Window Override	A logical zero written to this bit, enables standard color key operation. A logical one written to this bit enables video based only on the video window. Reset value is low (normal color key operation).
CR4	5	Set MODE0 State (CR37 = 1) (0) MODE0 pin low (1) MODE0 pin high	When CR37 = 1, this bit controls the state of the MODE0 output. A logical one written to this bit sets the MODE0 pin to high. A logical zero written to this bit sets the MODE0 pin low.
		Get MODE0 State (CR37 = 0) (0) MODE0 pin externally driven low (1) MODE0 pin externally driven high	When CR37 = 0, this bit indicates the state of the MODE0 input. A logical one read from this bit indicates that the MODE0 pin is driven high. A logical zero read from this bit indicates that the MODE0 pin is driven high. A logical zero read from this bit indicates that the MODE0 pin is driven low.
CR4	4	Video Pixel Display Order (0) Little Endian (1) Big Endian	This bit controls the display order of the video pixel port. A logical zero written to this bit displays video pixels in least significant to most significant order. A logical one written to this bit displays video pixels in most significant to least significant order. Reset value is low (little endian order).
CR4	3	Color Key Mode Select (0) Before Palette (1) After Palette	This bit controls whether color key matching occurs on the pixel value before or after the palette. A logical zero written to this bit selects color key matching on the pixel value before the palette. A logical one written to this bit selects color key matching on the 24-bit RGB value after the palette. Reset value is low (color key before palette).
CR4	2	VideoCache <sup>™</sup> Unload Select (0) Unload Within Video Window (1) Unload From Start of Active Graphics Enable	This bit controls whether VideoCache <sup>™</sup> FIFO data is unloaded only within the video window or at all times during active graphics enable.
CR4	1.0	24-bit Component Order (00) RGB (01) BRG (10) BGR (11) Reserved	This bit controls the component latching order in 24-bit-per-pixel graphic modes.

## Internal Registers (continued)

**Pixel Read Mask Register**

The 8-bit pixel read mask register may be written to or read by the MPU at any time, and is not initialized at power-up. D0 is the least significant bit. The contents of this register are bit-wise ANDed with the pixel data prior to addressing the color palette RAM.

**Status Registers 1-2**

These two 8-bit status registers are provided for device identification and to monitor certain device states. They may be read by the MPU at any time. MPU write cycles to status register 1 are ignored. D0 is the least significant bit corresponding to SR10 or SR20. These registers are not reset during power-up/reset.

SR1	7-6	Chip Identification	These bits are identification values; SR17 = 1 and SR16 = 0.
SR1	5-4	Chip Revision	These bits are revision values; they are always logical zero (00).
SR1	3	Monitor Sense	This is the SENSE® bit. If it is a logical zero, one or more of the IOR, IOG, and IOB outputs have exceeded the internal voltage reference level (335 mV). This bit is used to determine the presence of a CRT monitor and, with diagnostic code, the difference between a loaded or unloaded RGB line can be discerned. The 360 mV reference has a ±100 mV tolerance when an external voltage reference equal to 1.235 V is used. A greater tolerance is expected when an internal reference equal to 1.2 V is used.
SR1	2	Read/Write access status. (0) Write Cycle (1) Read Cycle	This bit provides RD/WR status when Address Register 0x00, 0x03, 0x04, or 0x07 has been written. When Address Register 0x00 or 0x04 has been written, the device is in the write mode and this bit is a logical zero. When address register 0x03 or 0x07 has been written, the device is in the read mode and this bit is a logical one.
SR1	1-0	RGB Component Counter (00) Red Color Component (01) Green Color Component (10) Blue Color Component	When read, these bits reflect the color component address for the next RD/WR cycle when accessing the palette, cursor color registers, or overscan register.
SR2	7	VideoCache™ FIFO Underflow	Reading this bit as a one indicates that VideoCache™ FIFO underflow occurred. Reset by writing any value to Status Register 2.
SR2	6-0	Reserved	These bits will always be read as zero.

## Internal Registers (continued)

**Video Window XSTART**

Video Window XSTART is a 12-bit register that stores the starting X position on the screen for a video window. A value of zero indicates that the video window begins in the first (leftmost) pixel of each horizontal scan line.

**Video Window YSTART**

Video Window YSTART is a 12-bit register that stores the starting Y position on the screen for a video window. A value of zero indicates that the video window begins on the first active graphics scan line.

**Video Window XWIDTH**

Video Window XWIDTH is a 12-bit register that stores the number of pixels per scan line within the video window. A value of zero indicates that no pixels are in the video window.

**Video Window YHEIGHT**

Video Window YHEIGHT is a 12-bit register that stores the number of scan lines within the video window. A value of zero indicates that no scan lines are within the video window.

**XSCALEINIT**

XDDAINTT is a 12-bit register that stores the initial term for the horizontal scaler.

**XSCALEINC**

XSCALEINC is a 12-bit register that stores the increment term for the horizontal scaler.

**Serial Clock Enable Start  
(Horizontal and Vertical)**

Serial clock enable start (horizontal and vertical) are 12-bit registers that store the number of scan lines and DIVCLK1 cycles before enabling the external clock gate, starting at the leading edge of HSYNC<sup>®</sup> for the horizontal direction and the leading edge of VSYNC<sup>®</sup> for the vertical direction.

**Serial Clock Enable Duration  
(Horizontal and Vertical)**

Serial clock enable duration (horizontal and vertical) are 12-bit registers that store the number of serial shift clock cycles to be generated per scan line in units of DIVCLK1 cycles for the horizontal direction, and in units of scan lines for the vertical direction.

**DIVCLK1 and DIVCLK2 Rate**

DIVCLK1 and DIVCLK2 rate are two 3-bit registers that control the divide rate of the free running DIVCLK1 and DIVCLK2 divide-down clocks, respectively. The divide-down ratios need not be the same as the input mux rate:

- (000) - 1:1
- (001) - 2:1
- (010) - 4:1
- (011) - 8:1
- (100-111) - Reserved

**VideoCache™ FIFO Size**

This register indicates the length of the VideoCache™ FIFO buffer in 16-byte units. This is a read-only register.

## Internal Registers (continued)

## Cursor (x,y) Registers

These registers are used to specify the (x,y) coordinate of the 64 x 64 x 2 hardware cursor. The cursor (x) register is made up of the cursor (x) low register (CXLR) and the cursor (x) high register (CXHR); the cursor (y) register is made up of the cursor (y) low register (CYLR) and the cursor (y) high register (CYHR). The last value written by the MPU to these registers is the value returned on a read. These registers may be written to or read by the MPU at any time.

CXLR and CXHR are cascaded to form a 12-bit cursor (x) register. Similarly, CYLR and CYHR are cascaded to form a 12-bit cursor (y) register. Bits D4-D7 of CXHR and CYHR are ignored and should be written as zeros.

The cursor (x) value to be written is calculated as follows:

$$X_p = \text{desired display screen (x) position} + 64$$

where the (x) reference point for the display screen,  $x = 0$ , is the upper left corner of the screen. The  $X_p$  position

equation places the upper lefthand corner of the cursor RAM array to the desired screen location. This allows the cursor position to be defined in the same coordinate space as the screen.

Values from 0 to 4095 may be written into the cursor (x) register. If  $X_p$  is equal to zero, the cursor will be entirely offscreen.

The cursor (y) value to be written is calculated as follows:

$$Y_p = \text{desired display screen (y) position} + 64$$

where the (y) reference point for the display screen,  $y = 0$ , is the upper left corner of the screen. The  $Y_p$  position equation places the upper left corner of the cursor RAM array to the desired screen location. This allows the cursor position to be defined in the same coordinate space as the screen.

Values from 0 to 4095 may be written into the cursor (y) register. If  $Y_p$  is equal to zero, the cursor will be entirely offscreen (see Cursor Operation in the Circuit Description section).

	Cursor (x) High (CXHR)				Cursor (x) Low (CXLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Address	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

	Cursor (y) High (CYHR)				Cursor (y) Low (CYLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Address	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0



## Internal Registers (continued)

Register Name	Initial Value
Command Register 0	0
Command Register 1	0
Command Register 2	0
Command Register 3	0
Command Register 4	0
Video Window XSTART-Low & High	Not Initialized
Video Window YSTART-Low & High	Not Initialized
Video Window XWIDTH-Low & High	Not Initialized
Video Window YHEIGHT-Low & High	Not Initialized
XSCALEINT-Low & High	Not Initialized
XSCALEINC-Low & High	Not Initialized
Serial Clock Enable Start (Horizontal)-Low & High	Not Initialized
Serial Clock Enable Duration (Horizontal)-Low & High	Not Initialized
Serial Clock Enable Start (Vertical)-Low & High	Not Initialized
Serial Clock Enable Duration (Vertical)-Low & High	Not Initialized
DIVCLK1 Rate	0
DIVCLK2 Rate	0
Color Mask	0
Color Key	0
FIFO Size	0x32
Color Palette RAM	Not Initialized
Pixel Read Mask	0xFF
Cursor Colors	Not Initialized
Overscan Color	Not Initialized
Cursor X,Y	Not Initialized
Cursor RAM array	Not Initialized

Table 13: Register Values on Reset.

## Pin Descriptions

Pin Name	I/O	Pins	Description												
RESET*	I	72	Reset input (TTL compatible). When this signal is low, all the command register bits are initialized to zero and the device is in VGA mode.												
BLANK*	I	98	Composite blank control input (TTL compatible). A logical zero drives the analog outputs to the blanking level, as specified in Tables 8 and 9. It is latched on the rising edge of GLCLK. When BLANK* is a logical zero, the pixel inputs are ignored. The falling edge of this signal determines the polarity of the HSYNC* and VSYNC* input pins. The onboard cursor positioning counters are referenced to this signal.												
ENABLE	I	96	Composite display enable control input (TTL compatible). The state of this signal and BLANK* determines whether the analog outputs are blanked or contain cursor color, pixel, or overscan data. This signal is latched on the rising edge of GLCLK. If overscanning is not used, this pin should be tied to BLANK*. The following table lists the combinations of ENABLE and BLANK*:												
			<table><tr><th>ENABLE</th><th>BLANK*</th><th>Operation</th></tr><tr><td>x</td><td>0</td><td>Video Blanking</td></tr><tr><td>0</td><td>1</td><td>Overscan Data</td></tr><tr><td>1</td><td>1</td><td>Cursor Color or Pixel Data</td></tr></table>	ENABLE	BLANK*	Operation	x	0	Video Blanking	0	1	Overscan Data	1	1	Cursor Color or Pixel Data
ENABLE	BLANK*	Operation													
x	0	Video Blanking													
0	1	Overscan Data													
1	1	Cursor Color or Pixel Data													
ODD/EVEN*	I	95	Odd/even field input (TTL compatible). This signal should be changed only during vertical blank. This input is used to ensure proper operation of the onboard cursor when interlaced operation (command bit CR23 = 1) is selected. When this signal is a logical zero, an even field is specified. When this signal is a logical one, an odd field is specified. This input is ignored if noninterlaced operation (command bit CR23 = 0) is selected.												
OSC, OSC*	I	131, 132	Pixel clock input (ECL/TTL compatible). This input is an ECL-compatible input, but a TTL clock may be used on either OSC or OSC* if selected by CR24 in Command Register 1 (CR34 = 0). It is recommended that all clock inputs be driven by a dedicated buffer to avoid reflection-induced jitter.												
DIVCLK1	O	127	Frame buffer shift clock output (TTL compatible). The signal on this pin is equal to the selected pixel clock divided by 8, 4, 2, or 1, depending on the selection in the DIVCLK1 rate register. This output has low drive capability.												
DIVCLK2	O	128	Frame buffer shift clock output (TTL compatible). The signal on this pin is equal to the selected pixel clock divided by 8, 4, 2, or 1, depending on the selection in the DIVCLK2 rate register. This output has low drive capability.												
FIFO RESET*	I	14	A low value applied to this pin enables normal VideoCache™ FIFO operation. A transition from high to low on this pin resets the VideoCache™ FIFO.												
GLCLK	I	130	Graphics port input load clock (TTL compatible with hysteresis). The rising edge of this signal latches P7:0 (A-D), BLANK*, ENABLE, HSYNC*, and VSYNC*.												
VLCLK	I	13	Video port input load clock (TTL compatible with hysteresis). The rising edge of this signal latches P7:0 (E-H).												

## Pin Descriptions (continued)

Pin Name	I/O	Pins	Description
P7:0 (A-H)	I	See Diag	Pixel port inputs (TTL compatible). This port can be used in various modes as shown in Tables 10 and 11, for video and or graphics input. It is recommended that unused pins be tied to ground to lower the device's power consumption.
VALID	I	15	Video port input pixel data valid signal (TTL compatible).
READY	O	12	Video port input pixel data ready signal (TTL compatible, low drive). This signal can be synchronously sampled using the rising edge of VLCLK. This signal changes only following a rising edge of VLCLK.
HFULL	O	11	VideoCache™ FIFO half-full or greater signal. (TTL compatible, low drive).
SEN	O	126	DIVCLK1 gating control signal (TTL compatible, low drive). It may be used to externally gate the DIVCLK1 output to generate a gated version of DIVCLK1. This signal changes only during DIVCLK1 low duration. The start time and duration of the pulse train may be programmed relative to the leading edge of HSYNC* and VSYNC*.
MODE0	I/O	91	General purpose registered input/output (TTL compatible) set or read using CR45. Selection of input or output is made using CR37.
MODE1-3	I/O	29, 17, 19	Reserved for future expansion. Must be tied high with a 10K pullup resistor.
WR*	I	73	Write control input (TTL compatible with hysteresis). D0-D7 data is latched on the rising edge of WR*, and RS0-RS3 are latched on the falling edge of WR* during MPU write operations. RD* and WR* should not be asserted simultaneously.
RD*	I	74	Read control input (TTL compatible with hysteresis). To read data from the device, RD* must be a logical zero. RS0-RS3 are latched on the falling edge of RD* during MPU read operations. RD* and WR* should not be asserted simultaneously.
RS0-RS3	I	75- 78	Register select inputs (TTL compatible). RS0-RS3 specify the type of read or write operation being performed, as specified in Tables 1 and 2.
D0-D7	I/O	83- 90	Data bus (TTL compatible). Data is transferred into and out of the device over this 8-bit bidirectional data bus. D0 is the least significant bit.
SENSE*	O	71	Comparator sense output (CMOS compatible). This pin will be low if one or more of the IOR, IOG, and IOB analog output levels is above the internal comparator reference of 350 mV $\pm$ 50 mV.
IOR, IOG, IOB	A,O	55, 58, 61	Red, green, and blue current outputs. These high-impedance current sources can directly drive a doubly-terminated 75 $\Omega$ coaxial cable (see the PC Board Layout Considerations section for further information).
HSYNC*	I	97	Horizontal sync control input (TTL compatible).
VSYNC*	I	99	Vertical sync control input (TTL compatible).

## Pin Descriptions (continued)

Pin Name	I/O	Pins	Description																			
HSYNCOUT <sup>™</sup> VSYNCOUT <sup>™</sup>	O	48, 49	Pipeline delayed horizontal and vertical sync control signals.																			
FSADJUST	A,I	52	<p>Full-scale adjust control. The IRE relationships in Figures 4 and 5 are maintained, regardless of the full-scale output current.</p> <p>When an external or the internal voltage reference is used (see Figures 10 and 11 in the PC Board Layout Considerations section), a resistor (RSET) connected between this pin and GND controls the magnitude of the full-scale video signal. The relationship between RSET and the full-scale output current on each output is:</p> $RSET (\Omega) = K \cdot 1,000 \cdot VREF (V) / I_{out} (mA)$ <p>K is defined in the table below. It is recommended that a 147 <math>\Omega</math> RSET resistor be used for doubly-terminated 75 <math>\Omega</math> loads (i.e., RS-343A applications).</p> <table><tr><th rowspan="2">Setup</th><th colspan="2">Sync Enabled</th><th colspan="2">Sync Disabled</th></tr><tr><th>0 IRE</th><th>7.5 IRE</th><th>0 IRE</th><th>7.5 IRE</th></tr><tr><td>K (8-bit)</td><td>2.888</td><td>3.055</td><td>2.045</td><td>2.207</td></tr><tr><td>K (6-bit)</td><td>3.000</td><td>3.170</td><td>2.100</td><td>2.260</td></tr></table> <p>K values are subject to change upon completion of characterization.</p>	Setup	Sync Enabled		Sync Disabled		0 IRE	7.5 IRE	0 IRE	7.5 IRE	K (8-bit)	2.888	3.055	2.045	2.207	K (6-bit)	3.000	3.170	2.100	2.260
Setup	Sync Enabled		Sync Disabled																			
	0 IRE	7.5 IRE	0 IRE	7.5 IRE																		
K (8-bit)	2.888	3.055	2.045	2.207																		
K (6-bit)	3.000	3.170	2.100	2.260																		
VREF OUT	A,O	68	Voltage reference output. This output provides a 1.2 V (typical) reference and may be connected directly to the VREF pin. If the on-chip reference is not used, this pin may be left floating. See Figures 10 and 11. Up to four Bt885s can be driven by this output.																			
VREF IN	A,I	67	Voltage reference input. If an external voltage reference is used (Figure 11), it must supply this input with a 1.2 V (typical) reference. A 0.1 $\mu$ F ceramic capacitor must be used to decouple this input to GND, as shown in Figures 10 and 11. The decoupling capacitor must be as close to the device as possible to keep lead lengths to an absolute minimum. When the internal reference is used, this pin should not drive any external circuitry, other than the decoupling capacitor (Figure 10).																			
COMP	A,O	64	Compensation pin. A 0.1 $\mu$ F ceramic capacitor must be used to bypass this pin to VAA. The COMP capacitor must be as close as possible to the device to keep lead lengths to an absolute minimum.																			
VAA	A,P	See Diag	Analog power. All VAA pins must be connected to the same analog power plane.																			
AGND	G	See Diag	Analog ground. All AGND pins must be connected to a common ground plane.																			
GND	G	See Diag	Digital ground. All GND pins must be connected to a common ground plane.																			



**Note:** All pins marked NC are reserved for future expansion and **MUST** be left floating.

## PC Board Layout Considerations

### PC Board Considerations

For optimum performance of the Bt885, proper CMOS CacheDAC™ layout techniques should be studied in the Bt451/7/8 Evaluation Module Operation and Measurements, Application Note (AN-16), before PC board layout is begun. This application note can be found in Brooktree's *Applications Handbook*.

The layout should be optimized for lowest noise on the Bt power and ground planes by providing good decoupling. The trace length between groups of VAA and GND pins should be as short as possible to minimize inductive ringing.

A well-designed power distribution network is critical to eliminating digital switching noise. The ground plane must provide a low-impedance return path for the digital circuits. A PC board with a minimum of four layers is recommended, with layers 1 (top) and 4 (bottom) for signals and layers 2 and 3 for power and ground.

### Component Placement

Components should be placed as close as possible to the associated CacheDAC™ pin. Whenever possible, components should be placed so traces can be connected point to point.

The optimum layout enables the Bt885 to be located as close as possible to the power supply connector and the video output connector.

### Ground Planes

For optimum performance, a common digital and analog ground plane is recommended.

### Power Planes

Separate digital and analog power planes are recommended. The digital power plane should provide power to all digital logic on the PC board, and the analog power plane should provide power to all Bt885 power pins, VREF circuitry, and COMP and VREF decoupling. There should be at least a 1/8-inch gap between the digital power plane and the analog power plane.

The analog power plane should be connected to the digital power plane (VCC) at a single point through a ferrite bead, as illustrated in Figures 10 and 11. This bead should be located within 3 inches of the Bt885. The bead provides resistance to switching currents, acting as a resistance at high frequencies. A low-resis-

tance bead should be used, such as Ferroxcube 5659065-3B, Fair-Rite 2743001111, or TDK BF45-4001.

### Device Decoupling

For optimum performance, all capacitors should be located as close as possible to the device, and the shortest possible leads (consistent with reliable operation) should be used to reduce the lead inductance. Chip capacitors are recommended for minimum lead inductance. Radial lead ceramic capacitors may be substituted for chip capacitors and are better than axial lead capacitors for self-resonance. Values are chosen to have self-resonance above the pixel clock.

### Power Supply Decoupling

The best power supply decoupling performance is obtained with a 0.1  $\mu\text{F}$  ceramic capacitor in parallel with a 0.01  $\mu\text{F}$  chip capacitor decoupling each group of VAA pins to GND. The capacitors should be placed as close as possible to the device VAA and GND pins and connected with short, wide traces.

The 10  $\mu\text{F}$  capacitor shown in Figures 10 and 11 is for low-frequency power supply ripple; the 0.1  $\mu\text{F}$  capacitors are for high-frequency power supply noise rejection.

When a linear regulator is used, the power-up sequence must be verified to prevent latchup. A linear regulator is recommended to filter the analog power supply if the power supply noise is greater than or equal to 200 mV. This is especially important when a switching power supply is used, and the switching frequency is close to the raster scan frequency. About 10% of the power supply hum and ripple noise less than 1 MHz will couple onto the analog outputs.

### COMP Decoupling

The COMP pin must be decoupled to VAA, typically with a 0.1  $\mu\text{F}$  ceramic capacitor. Low-frequency supply noise will require a larger value. The COMP capacitor must be as close as possible to the COMP and VAA pins. A surface-mount ceramic chip capacitor is preferred for minimal lead inductance. Lead inductance degrades the noise rejection of the circuit. Short, wide traces will also reduce lead inductance.

If the display has a ghosting problem, additional capacitance in parallel with the COMP capacitor may help.

## PC Board Layout Considerations (continued)

### VREF Decoupling

A 0.1  $\mu\text{F}$  ceramic capacitor should be used to decouple this input to GND.

### Digital Signal Interconnect

The digital inputs to the Bt885 should be isolated as much as possible from the analog outputs and other analog circuitry. Also, these input signals should not overlay the analog power plane or analog output signals.

Most of the noise on the analog outputs will be caused by excessive edge rates (less than 3 ns), overshoot, undershoot, and ringing on the digital inputs.

The digital edge rates should not be faster than necessary, as feedthrough noise is proportional to the digital edge rates. Lower-speed applications will benefit from using lower-speed logic (3–5 ns edge rates) to reduce data-related noise on the analog outputs.

Transmission lines will mismatch if the lines do not match the source and destination impedance. This will degrade signal fidelity if the line length reflection time is greater than one fourth the signal edge time (refer to Brooktree Application Notes AN-11 and AN-12). Line termination or line-length reduction is the solution. For example, logic edge rates of 2 ns require line lengths of less than 4 inches without use of termination. Ringing may be reduced by damping the line with a series resistor (30–300  $\Omega$ ). The RS-select inputs and RD\*/WR\* lines must be verified for proper levels with no ringing, undershoot, or overshoot. Ringing on these lines can cause improper operation.

Radiation of digital signals can also be picked up by the analog circuitry. This is prevented by reducing the digital edge rates (rise/fall time), minimizing ringing with damping resistors, and minimizing coupling through PC board capacitance by routing the digital signals at a 90 degree angle to any analog signals.

The clock driver and all other digital devices must be adequately decoupled to prevent noise generated by the digital devices from coupling into the analog circuitry.

### Clock Interfacing

The Bt885 requires a pixel clock with monotonic clock edges for proper operation. Impedance mismatch on the pixel clock line will induce reflections on the pixel clock, which may cause erratic operation.

The Pixel Clock Pulse Width High Time and Pixel Clock Pulse Width Low Time minimum specifications (see the AC Characteristics section) must not be violated, or erratic operation can occur.

The pixel clock line must be terminated to prevent impedance mismatch. A series termination of 33–68  $\Omega$  placed at the pixel clock driver may be used, or a parallel termination may be used at the pixel clock input to the CacheDAC™. A parallel termination of 220  $\Omega$  to VCC and 330  $\Omega$  to ground will provide a Thevenin equivalent of a 110  $\Omega$  termination, which is normally sufficient to absorb reflections. The series or parallel resistor values should be adjusted to provide the optimum clock signal fidelity.

### MPU Control Signal Interfacing

The Bt885 uses the RD\*, WR\*, and RS lines to determine which MPU accesses will take place. Glitches or ringing on any of these lines may cause improper MPU operation. When a VGA controller with edge rate control is used on these lines, a series termination is not necessary. In non-VGA controller application or in applications where the MPU control signals are daisy chained, a series termination, pull-down resistors, or additional capacitance to ground should be used to prevent glitches that could cause improper MPU accesses.

### Analog Signal Interconnect

The Bt885 should be located as close as possible to the output connectors to minimize noise pickup and reflections caused by impedance mismatch.

The analog outputs are susceptible to crosstalk from digital lines; digital traces must not be routed under or adjacent to the analog output traces.

To maximize the high-frequency power supply rejection, the video output signals should not overlay the analog power plane.

For maximum performance, the analog video output impedance, cable impedance, and load impedance should be the same. The load resistor connection between the video outputs and GND should be as close as possible to the Bt885 to minimize reflections. Unused analog outputs should be connected to GND.

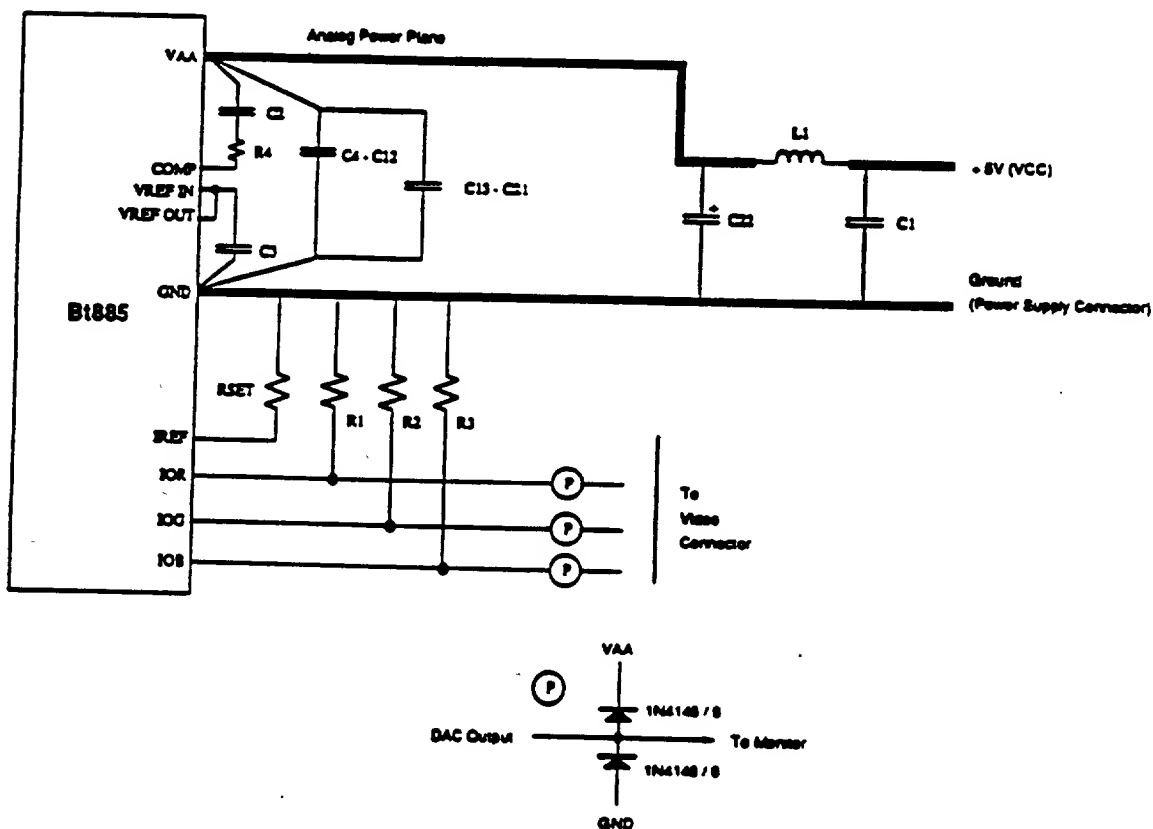
Analog output video edges exceeding the CRT monitor bandwidth can be reflected, producing cable-length dependent ghosts. Simple pulse filters can reduce high-frequency energy, reducing EMI and noise. The filter impedance must match the line impedance.

### **Analog Output Protection**

The Bu885 analog output should be protected against high-energy discharges, such as those from monitor arc-over or from hot-switching AC-coupled monitors.

The diode protection circuit shown in Figures 10 and 11 can prevent latchup under severe discharge condi-

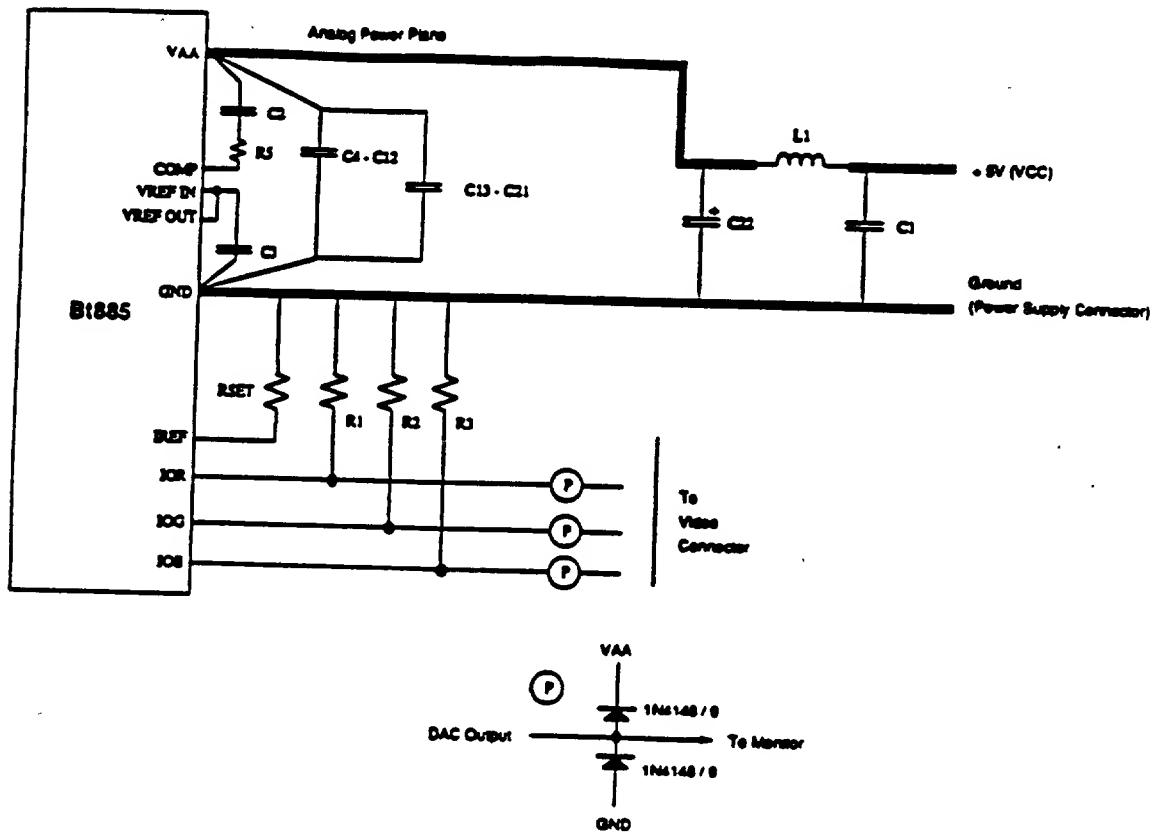
tions without adversely degrading analog transition times. The 1N4148/9 parts are low-capacitance, fast-switching diodes, which are also available in multiple-device packages (FSA250X or FSA270X) or surface-mountable pairs (BAV99 or MMBD7001).



Location	Description	Vendor Part Number
C1-C12	0.1 $\mu$ F ceramic capacitor	Erie RPE112Z5U104M50V
C13-C21	0.1 $\mu$ F ceramic chip capacitor	AUX12102T103QA1018
C22	10 $\mu$ F capacitor	Mallory CSR13G106KM
L1	ferrite bead	Fair-Rite 2743001111
R1, R2, R3	75 $\Omega$ 1% metal film resistor	Dale CMF-55C
R4	15 $\Omega$ 1% metal film resistor	Dale CMF-55C
RSET	1% metal film resistor	Dale CMF-55C

**Figure 10. Typical Connection Diagram and Parts List (Internal Voltage Reference).**





Location	Description	Vendor Part Number
C1-C12	0.1 $\mu$ F ceramic capacitor	Erie RPE112Z5U104M50V
C13-C21	0.1 $\mu$ F ceramic chip capacitor	AUX12102T103QA1018
C22	10 $\mu$ F capacitor	Mallory CSR13G106KM
L1	ferrite bead	Fair-Rite 2743001111
R1, R2, R3	75 $\Omega$ 1% metal film resistor	Dale CMF-55C
R4	15 $\Omega$ 1% metal film resistor	Dale CMF-55C
R5	15 $\Omega$ 1% metal film resistor	Dale CMF-55C
RSET	1% metal film resistor	Dale CMF-55C

Figure 11. Typical Connection Diagram and Parts List (External Voltage Reference).

## Application Information

### *Using Multiple Devices*

When multiple Bt885s are used, each Bt885 should have its own power plane ferrite bead. If the internal reference is used, each Bt885 should use its own internal reference.

Although the multiple Bt885s may be driven by a common external voltage/current reference, higher performance may be obtained if each CacheDAC™ uses its own reference. This will reduce the amount of color channel crosstalk and color palette interaction.

Each Bt885 must still have its own RSET resistor, analog output termination resistors, power supply bypass capacitors, COMP capacitor, and reference capacitors.

### *ESD and Latchup Considerations*

Correct ESD-sensitive handling procedures are required to prevent device damage, which can produce symptoms of catastrophic failure or erratic device behavior with leaky inputs.

All logic inputs should be held low until power to the device has settled to the specified tolerance. DAC power decoupling networks with large time constants should be avoided. They could delay VAA power to the device. Ferrite beads must be used only for analog

power VAA decoupling. Inductors cause a time constant delay that induces latchup.

Latchup can be prevented by ensuring that all VAA and GND pins are at the same potential and that the VAA supply voltage is applied before the signal pin voltages. The correct power-up sequence ensures that any signal pin voltage will never exceed the power supply voltage by more than +0.5 V.

### *Reference Selection*

An external voltage reference provides about ten times the power supply rejection on the analog outputs than does an external current reference.

### *Sleep Operation*

When the internal or external voltage reference is used, the DACs will be turned off during sleep mode.

When an external voltage reference is used, some internal circuitry will still be powered during the sleep mode, resulting in 0.5 mA of power supply current being drawn (above the rated supply current specifications). This unnecessary current drain can be disabled by turning off the external voltage reference during power-down mode.

## Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Power Supply	VAA	4.75	5.00	5.25	V
Ambient Operating Temperature	TA	0		+70	°C
Output Load	RL		37.5		$\Omega$
Voltage Reference Configuration Reference Voltage	VREF	1.1112	1.235	1.359	V

## Absolute Maximum Ratings

Parameter	Symbol	Min	Typ	Max	Units
VAA (measured to GND)				7.0	V
Voltage on Any Signal Pin (Note 1)		GND-0.5		VAA + 0.5	V
Analog Output Short Circuit Duration to Any Power Supply or Common	ISC		indefinite		
Storage Temperature	TS	-65		+150	°C
Junction Temperature	TJ			+150	°C
Vapor Phase Soldering (1 minute)	TVSOL			220	°C

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those listed in the operational sections of this specification are not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

*Note 1:* This device employs high-impedance CMOS devices on all signal pins. It should be handled as an ESD-sensitive device. Voltage on any signal pin that exceeds the power supply voltage by more than +0.5 V can cause destructive latchup.

## DC Characteristics

Parameter	Symbol	Min	Typ	Max	Units
Resolution (each DAC)		8	8	8	
Accuracy (each DAC)					
Integral Linearity Error	IL			±1	LSB
Differential Linearity Error	DL			±1	LSB
Gray-Scale Error			Guaranteed	±5	% Gray
Monotonicity					Scale
Coding					Binary
Digital Input					
Input High Voltage	V <sub>IH</sub>	2.0		V <sub>AA</sub> + 0.5	V
Input Low Voltage	V <sub>IL</sub>	GND-0.5		0.8	V
Input High Current (V <sub>in</sub> = 2.4 V)	I <sub>IH</sub>			1	μA
Input Low Current (V <sub>in</sub> = 0.4 V)	I <sub>IL</sub>			-1	μA
Input Capacitance	C <sub>IN</sub>			7	pF
(v = 1 MHz, V <sub>in</sub> = 2.4 V)					
Hysteresis			0.3		V
Digital Output					
Output High Voltage	V <sub>OH</sub>	2.4			V
(I <sub>OH</sub> = -400 μA)					
Output Low Voltage	V <sub>OL</sub>			0.4	V
(I <sub>OL</sub> = 3.2 mA)					
Three-State Current	I <sub>OZ</sub>			50	μA
Output Capacitance	C <sub>DOUT</sub>			7	pF
Load Capacitance	C <sub>L</sub>			10	pF

See test conditions on next page.

## DC Characteristics (continued)

Parameter	Symbol	Min	Typ	Max	Units
Analog Output					
Gray-Scale Current Range				20	mA
Output Current (Standard RS-343A)					
White Level Relative to Black		15.86	17.62	18.5	mA
Black Level Relative to Blank					
SETUP = 7.5 IRE		0.95	1.44	1.90	mA
SETUP = 0 IRE		0	5	50	μA
Blank Level		6.29	7.62	8.96	mA
Sync Level		0	5	50	μA
LSB Size			69.1		μA
DAC-to-DAC Matching			2	5	%
Output Compliance	VOC	-1.0		+1.5	V
Output Impedance	RAOUT		10		kΩ
Output Capacitance	CAOUT			30	pF
(f = 1 MHz, IOUT = 0 mA)					
Onboard VREF (Note 1)	VREFOUT	TBD	TBD	TBD	V
Voltage Reference Input Current	IVR IN		tbd	tbd	mA
Power Supply Rejection Ratio	PSRR			0.5	% / % ΔVAA
(COMP = 0.1 μF, f = 1 KHz)					

Test conditions to generate RS-343A standard video signals (unless otherwise specified): "Recommended Operating Conditions" using external voltage reference with SETUP = 7.5 IRE, RSET = 147 Ω, VREF = 1.235 V. As the above parameters are guaranteed over the full temperature range, temperature coefficients are not specified or required.

When the internal voltage reference is used, RSET may require adjustment to meet these limits. Also, the "gray-scale" output current (white level relative to black) will have a typical tolerance of ±10% rather than the ±5% specified above.

When the device is in the 6-bit mode, the output levels are approximately 1.5% lower than these values.

Note 1: Onboard VREF numbers subject to change upon completion of characterization.

## AC Characteristics

Parameter	Symbol	135 MHz Devices			110 MHz Devices			Units
		Min	Typ	Max	Min	Typ	Max	
OSC, OSC* All Mux Rates	F <sub>max</sub>			135			110	MHz
RS0-RS3 Setup Time	1	10			10			ns
RS0-RS3 Hold Time	2	10			10			ns
RD* Asserted to D0-D7 Driven	3	2			2			ns
RD* Asserted to D0-D7 Valid	4			40			40	ns
RD* Negated to D0-D7 3-States	5			20			20	ns
Read D0-D7 Hold Time	6	2						ns
Write D0-D7 Setup Time	7	10			10			ns
Write D0-D7 Hold Time	8	10			10			ns
RD*, WR* Pulse Width Low	9	50			50			ns
RD*, WR* Pulse Width High	10	6*pc <sub>clk</sub>			6*pc <sub>clk</sub>			ns
GLCLK Rates	G <sub>max</sub>							
8:1 Multiplexing				16.9			13.75	MHz
4:1 Multiplexing				33.8			27.5	MHz
2:1 Multiplexing				67.5			55	MHz
1:1 Multiplexing				90			90	MHz
VLCLK Rate	V <sub>max</sub>			85			85	MHz
DIVCLK1, DIVCLK2 Rates	D <sub>max</sub>			67.5			55	MHz
OSC, OSC* Cycle Time (Note 1)	11	14.81			18.18			ns
All Mux Rates								
OSC, OSC* Pulse Width High	12				tbd			ns
All Mux Rates								
OSC, OSC* Pulse Width Low	13				tbd			ns
All Mux Rates								
Duty Cycle of Selected Pixel Clock When Clock Doubler Enabled		45		55	45		55	%
GLCLK Cycle Time	14							
8:1 Multiplexing		59.17			72.72			ns
4:1 Multiplexing		29.58			36.36			ns
2:1 Multiplexing		14.81			18.18			ns
1:1 Multiplexing		11.11			11.11			ns
GLCLK Pulse Width High	15							
8:1 Multiplexing		4			4			ns
4:1 Multiplexing		4			4			ns
2:1 Multiplexing		4			4			ns
1:1 Multiplexing		4			4			ns
GLCLK Pulse Width Low	16							
8:1 Multiplexing		4			4			ns
4:1 Multiplexing		4			4			ns
2:1 Multiplexing		4			4			ns
1:1 Multiplexing		4			4			ns

Test conditions at end of this section.

## AC Characteristics (continued)

Parameter	Symbol	135 MHz Devices			110 MHz Devices			Units
		Min	Typ	Max	Min	Typ	Max	
VLCLK Cycle Time	17	11.76			11.76			ns
VLCLK Pulse Width High	18	4			4			ns
VLCLK Pulse Width Low	19	4			4			ns
DIVCLK1, DIVCLK2 Cycle Time	20	14.81			14.81			ns
DIVCLK1, DIVCLK2 Duty Cycle	21	40		60	40		60	%
Graphics Data Setup to GLCLK	22	3			3			ns
Graphics Data Hold from GLCLK	23	1			1			ns
Data Setup to GLCLK ENABLE, BLANK*, HSYNC*, VSYNC*	24	3			3			ns
Data Hold to GLCLK ENABLE, BLANK*, HSYNC*, VSYNC*	25	1			1			ns
Video Data Setup to VLCLK	26	3						ns
Video Data Hold from VLCLK	27	1						ns
VALID Setup to VLCLK	28	3			3			ns
VALID Hold from VLCLK	29	1			1			ns
VLCLK to READY Valid	30			7			7	ns
DIVCLK1 to SEN Valid	31			3			3	ns
FIFO Reset Pulse Width		10			10			ns
Analog Output Delay	32			30			30	ns
Analog Output Rise/Fall Time	33		3			3		ns
Analog Output Settling Time (Note 2)	34		13			13		ns
Clock and Data Feedthrough (Note 2)			-30			-30		dB
Glitch Impulse (Note 2)			75			75		pV - sec
SENSE* Output Delay	35		1			1		µs
DAC-to-DAC Crosstalk			-23			-23		dB
Analog Output Skew				2			2	ns
VAA Supply Current	IAA							mA
Normal Operation			tbd	tbd		tbd	tbd	mA
"Sleep" Mode (Note 3)			tbd	tbd		tbd	tbd	mA

Test conditions at end of this section.

## AC Characteristics (continued)

Pipeline Delay	
Graphics 1:1/No Video	3 LCLKS + 16 PCLKS
MUX Graphics/No Video (Note 1)	(8 LCLKS + 16 PCLKS) ± 2 LCLKS
Graphics 1:1/Video	27 LCLKS + 16 PCLKS
MUX Graphics/Video (Note 1)	(32 LCLKS + 16 PCLKS) ± 2 LCLKS
<i>Note 1:</i> The number of LCLKS will have to be multiplied by the respective MUX rates to get the proper number of pipeline delays. (i.e. the pipeline delay is 2:1 MUX Graphics/No video = (16 LCLKS + 16 PCLKS) ± 2 LCLKS).	

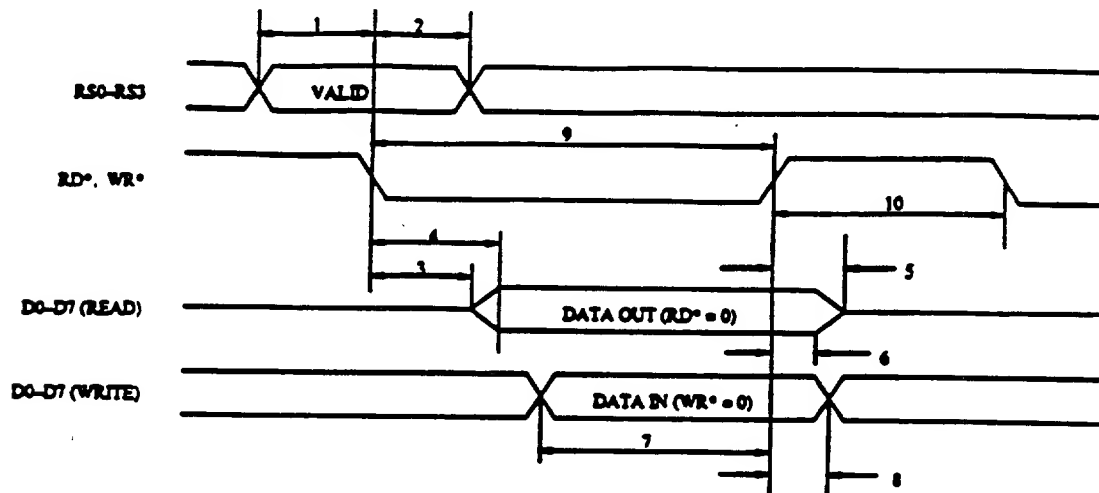
Test conditions (unless otherwise specified): "Recommended Operating Conditions" using external voltage reference with SETUP = 7.5 IRE, VREF = 1.235 V, RSET = 147Ω. TTL input values are 0–3 V, with input rise/fall times ≤ 3 ns, measured between the 10% and 90% points. Timing reference points at 50% for inputs and outputs. Analog output load ≤ 10 pF; SENSE\* and D0–D7 output load ≤ 50 pF. DIVCLK1, DIVCLK2 output load = 50 pF. As the above parameters are guaranteed over the full temperature range, temperature coefficients are not specified or required. Timing waveforms are shown in Figures 12–14.

*Note 1:* OSC and OSC\* cycle times assume the use of the 2Xclock Multiplier.

*Note 2:* Numbers guaranteed by design.

*Note 3:* External voltage reference is disabled during sleep mode, all inputs are low, and clock is running.



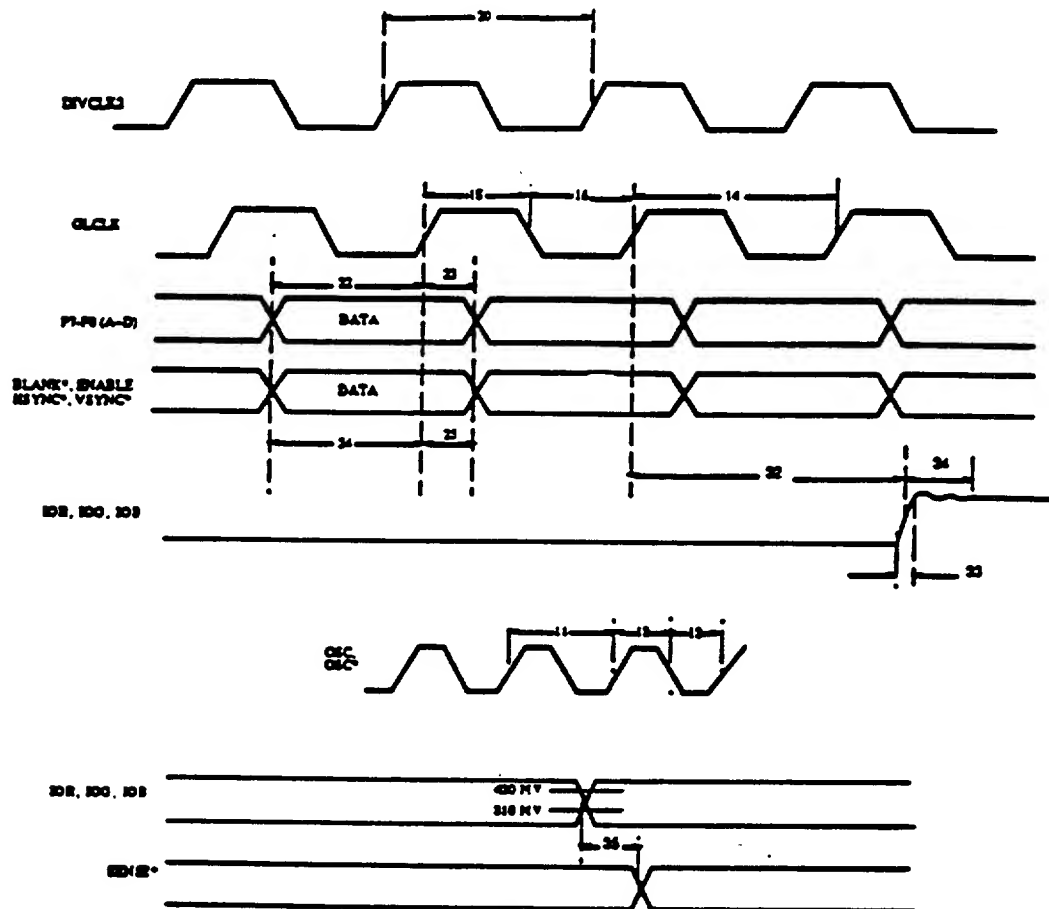


*Note 1:* Output delay measured from the 50% point of the rising edge of CLOCK to the 50% point of full-scale transition.

*Note 2:* Settling time measured from the 50% point of full-scale transition to the output remaining within  $\pm 1$  LSB.

*Note 3:* Output rise/fall time measured between the 10% and 90% points of full-scale transition.

**Figure 12. MPU Read/Write Timing.**



Note 1: Output delay measured from the 50% point of the rising edge of CLOCK to the 50% point of full-scale transition.

Note 2: Settling time measured from the 50% point of full-scale transition to the output remaining within  $\pm 1$  LSB.

Note 3: Output rise/fall time measured between the 10% and 90% points of full-scale transition.

Figure 13. Graphics Input/Output Timing

Timing Waveforms (continued)

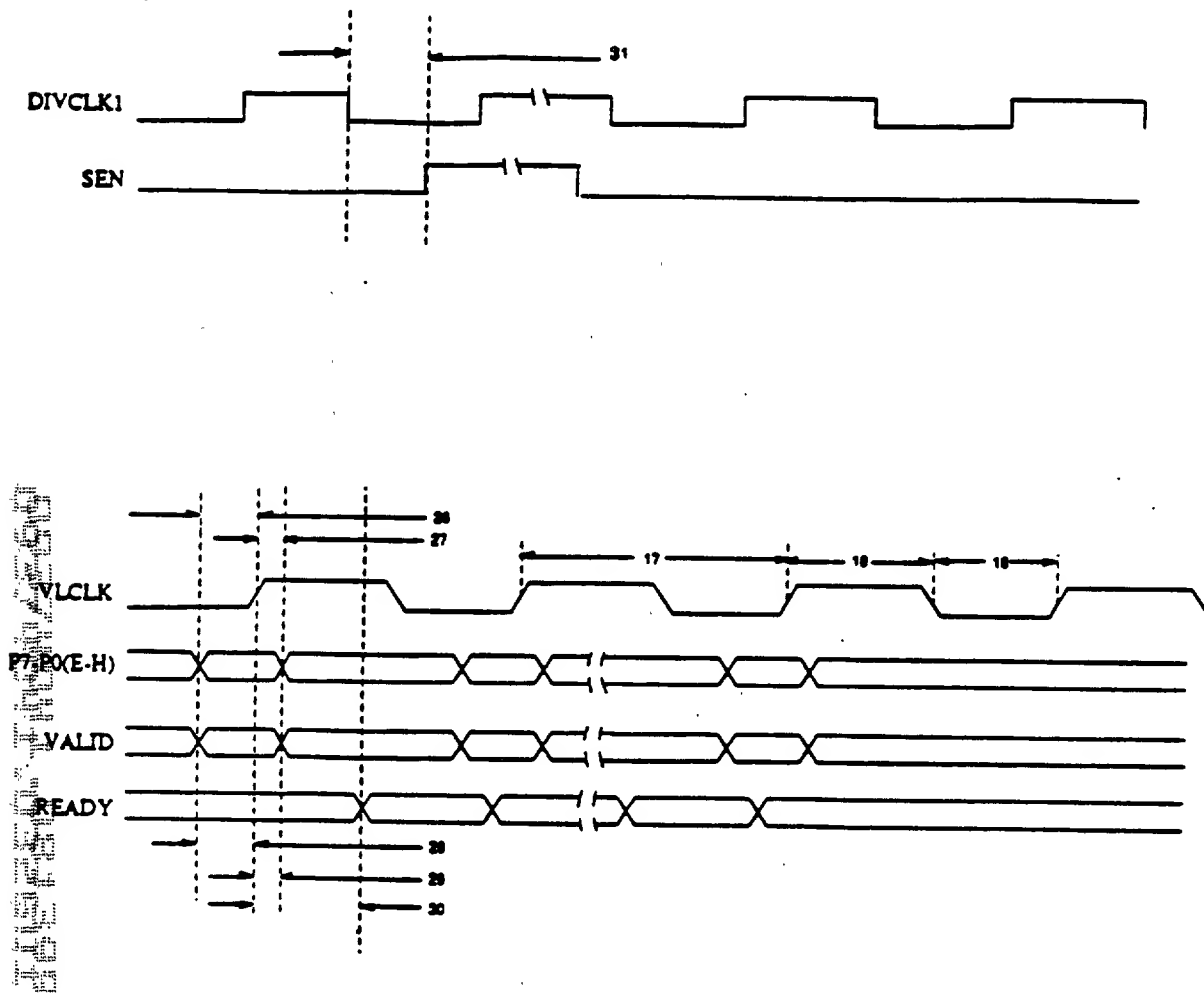


Figure 14. Video Input/Output Timing

**Bt885**

**Brooktree®**

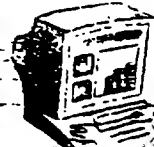
**Ordering Information**

Model Number	Speed	Package	Ambient Temperature Range
Bt885KHF135	135 MHz	160-pin Plastic Quad Flatpack	0° to +70°C
Bt885KHF110	110 MHz	160-pin Plastic Quad Flatpack	0° to +70°C

# 82750PD Video Processor Programmer's Reference Manual

EXHIBIT

EX-265



intel



ATI017761

# 82750PD Video Processor Programmer's Reference Manual

Order Number: 272352-001

ATI017762

In the United States, additional copies of this manual or other Intel literature may be obtained from

Intel Corporation  
Literature Sales  
P.O. Box 7641  
Mt. Prospect, IL 60056-7641  
1-800-548-4725

In locations outside the United States, obtain additional copies of Intel documentation by contacting your local Intel sales office.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication, or disclosure is subject to restrictions stated in Intel's software license agreements, or in the case of software delivered to the government, in accordance with the software license agreement as defined in FAR 52.227-7013.

No part of this document may be copied or reproduced in any form by any means, without prior written consent of Intel Corporation.

Intel Corporation retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specification before placing your order.

---

Revision	Date
-001	09/93

Copyright 1993 Intel Corporation  
All Rights Reserved

Intel Corporation  
2200 Mission College Drive  
Santa Clara, California 95052-8119

AT1017764



# Table of Contents

## Chapter 1 - Guide to this Manual

1.1 Manual Contents.....	1
1.2 Notational Conventions and Terminology .....	3

## Chapter 2 - Introduction to the 82750PD

2.1 Introduction .....	1
2.2 Memory and Registers .....	3
2.3 82750PD Memory Address Space.....	4
2.4 82750PD Core Registers.....	7
2.5 Shared Frame Buffer (SFB).....	7
2.5.1 Accessing the 64-Bit SFB .....	7
2.5.2 Accessing the 32 Bit SFB.....	8
2.5.3 Types of Memory on the SFB .....	9
2.6 Reserved Memory Locations .....	10

## Chapter 3 - 82750PD Core Architecture

3.1 Overview .....	1
3.2 General Purpose Registers .....	3
3.3 Arithmetic/Logic Unit (ALU).....	4
3.3.1 Condition Code Register .....	4
3.3.2 ALU Operations .....	5
3.3.3 82750PD Interrupts .....	7
3.3.4 Performance Monitoring .....	7
3.4 Barrel Shifter .....	7
3.5 Data RAM .....	8
3.6 Loop Counters .....	9
3.7 Microcode RAM .....	10

Contents-1

ATI017765

3.8	Horizontal Line Counter .....	12
3.9	Field Counter .....	12
3.10	Core Input FIFOs .....	13
3.11	Core Output FIFOs.....	18
3.12	Statistical Decoder .....	23
3.12.1	Statistical Codes .....	24
	Code Description Table .....	25
	Short Mode and End Mode .....	27
3.12.2	Statistical Decoder Control Register .....	30
3.12.3	Writing and Reading the Code Description Table .....	32
3.13	Pixel Interpolator .....	37
3.13.1	Sequential Interpolation .....	38
3.13.2	Random Interpolation.....	39
3.13.3	Pixel Interpolator Control .....	40
3.14	Control, Status, and Interrupt Flag Registers .....	43
3.14.1	Core Control Register.....	44
3.14.2	Core Interrupt Flag Register .....	46
3.14.3	Core Status Register .....	47
3.14.4	Summary of Interrupt Bits.....	49
3.15	Host Access to the Core Registers .....	50

## **Chapter 4 - Microcode Instruction Format**

---

4.1	Overview.....	1
4.2	Instruction Sequencing.....	1
4.3	Instruction Word Field Descriptions .....	2
4.3.1	NADDR -- Next Instruction Address Field .....	2
4.3.2	CFSEL -- Condition Flag Select Field .....	3
4.3.3	ASRC -- A Bus Source Select Field .....	5
4.3.4	ADST -- A Bus Destination Select Field.....	8
4.3.5	BSRC -- B Bus Source Select Field.....	8
4.3.6	BDST -- B Bus Destination.....	8
4.3.7	CNT -- Decrement Loop Counter Bit.....	8
4.3.8	LIT -- Literal Select Bit.....	9
4.3.9	SHFT -- Shift Control Field.....	9
4.3.10	ALUSS -- ALU Source Select Bits .....	10
4.3.11	ALUOP -- ALU Operation Code Field .....	10
4.3.12	LC -- LOOP Counter Select Bit .....	12

## Chapter 5 - UNIVERSAL HOST BUS INTERFACE

5.1 Introduction to the Universal Host Bus Interface.....	1
5.2 Host Interface Address Configuration .....	3
5.2.1 Configuration of Devices in a Single Slot.....	4
5.2.2 Address Configuration .....	5
ISA POS Information .....	7
EISA POS Register Format .....	7
Micro Channel POS Register Format.....	8
PCI POS Register Format.....	9
VL-Bus POS Information.....	10
5.3 Host I/O Registers.....	11
5.4 EMS-Style Memory Address Mode.....	14
5.4.1 "EMS" Memory Mapping and Registers.....	15
5.4.2 Setting Up the "EMS" Mode .....	17
POS Registers and the "EMS" Mode .....	18
"EMS" Configuration Registers.....	18
Page Address Registers.....	20
5.5 82750PD Indirect Access.....	21
5.6 Host-SFB FIFOs .....	25
5.6.1 Host-SFB Write FIFO .....	26
Basic Operation of the Host-SFB Write FIFO .....	28
Write FIFO Control Register (WFCNTR).....	29
Write FIFO Address Counter (WFCNTR) .....	34
Write FIFO Data Register (WFDATA) .....	35
Loading the Write FIFO Data Register .....	35
Saving and Restoring the State of the Write FIFO .....	37
Write FIFO Test Register .....	38
5.6.2 Host-SFB Read FIFO.....	39
Basic Operation of the Host-SFB Read FIFO.....	40
Read FIFO Control Register (RFCNTR) .....	41
Read FIFO Address Counter (RFCNTR) .....	45
Read FIFO Data Register (RFDATA).....	46
Read FIFO Test Register (RFTEST) .....	47
Reading the RFDATA Register .....	48
Saving and Restoring the State of the Read FIFO .....	50
5.7 Interrupts and Meta-Interrupts .....	50
5.7.1 The Interrupt/Meta-Interrupt System .....	51
5.7.2 General Control and General Status Registers .....	56
5.7.3 Setting Up the 82750PD Interrupt.....	57
5.8 Configuration Registers.....	58

5.8.2 I/O Base CFG .....	61
5.8.3 Bus CFG Register .....	62
5.8.4 Interrupt CFG Register .....	66
5.8.5 SFBI CFG Registers .....	68
5.8.6 "EMS" CFG Registers .....	70
5.8.7 Reserved CFG Registers .....	70

## Chapter 6 - SynchroLink

6.1 Introduction .....	1
6.2 Arbiter, Source, and Target .....	3
6.3 SynchroLink Data Packets .....	4
6.3.1 Arbiter, Source, and Target Interactions .....	4
6.3.2 Data Packet Fields .....	6
6.3.3 Service Requests .....	10
6.3.4 Service Completion Message .....	12
6.3.5 Broadcast Messages .....	14
6.4 SynchroLink Registers and Operations .....	16
6.4.1 SynchroLink Registers .....	16
6.4.2 Transceiver Registers (XCVRn) .....	18
6.4.3 Broadcast Transmit Mode .....	24
6.4.4 Broadcast Receive Mode .....	26
6.4.5 Sending a Request and Receiving a Completion .....	29
6.4.6 Receiving a Request and Sending a Completion .....	32
Receive Service Register (RSRV) .....	33
Operations .....	34
6.4.7 Message Status Register (MSTATUS) .....	36
6.4.8 Match Service Register (MSRV) .....	38
6.4.9 Configuring the SynchroLink Interface .....	39
Message Configuration Register (MCFG) .....	39
Configuration Procedure .....	42
6.5 Meta-Interrupts .....	43

## Appendix A - 82750PD Registers

## Appendix B - Logical and Device Addresses

## Appendix C - Programming Examples

C.1 Host-SFB FIFOs.....	1
C.1.1 Write FIFO.....	1
C.1.2 Read FIFO.....	2
C.2 Indirect I/O to a Configuration Register.....	4
C.3 Writing to the maddr Register.....	5
C.4 Writing to the pc Register.....	6

## Contents

## Figures

Figure 2-1. Shared Frame Buffer System.....	2
Figure 2-2. 82750PD Block Diagram.....	2
Figure 2-3. Block Diagram Showing Memory and Registers.....	3
Figure 3-1. 82750PD Core Block Diagram.....	2
Figure 3-2. Input FIFO Modes for Reading Data.....	17
Figure 3-3. Output FIFO Modes for Writing Data.....	21
Figure 3-4. SFB Bit-Stream Decoding Addresses.....	36
Figure 3-5. Pixel Interpolation.....	37
Figure 3-6. Sequential-2D Pixel Interpolation.....	38
Figure 3-7. Pixel Pair Phases.....	42
Figure 4-1. Microcode Instruction Format.....	1
Figure 4-2. Literal Field Mapping onto a Bus.....	9
Figure 5-1. Multiple Devices in a Single Logical Slot.....	4
Figure 5-2. Host I/O Registers.....	12
Figure 5-3. "EMS" Window Mapping.....	15
Figure 5-4. Details of "EMS" Mapping.....	16
Figure 5-5. 82750PD Indirect Register Mapping.....	22
Figure 5-6. 82750PD Indirect Register Address Translation.....	25
Figure 5-7. Block Diagram of the Read and Write FIFOs.....	25
Figure 5-8. Block Diagram of the Write FIFO.....	27
Figure 5-9. Data Paths from the Write FIFO Data Register to the SFB.....	36
Figure 5-10. Block Diagram of the Read FIFO.....	40
Figure 5-11. Data Paths from the SFB to the Read FIFO Data Register.....	49
Figure 5-12. Diagram of the Meta-Interrupt System.....	53
Figure 6-1. The SynchroLink with Devices Connected.....	2
Figure 6-2. Data Packet - Source Accepts.....	4

Figure 6-3. Data Packet - Source Accepts and Target Acknowledges .....	5
Figure 6-4. Data Stream - Invitation Not Acknowledged .....	6
Figure 6-5. Data Packet Format .....	7
Figure 6-6. Broadcast Transmit State Diagram .....	24
Figure 6-7. Send Service Request State Diagram .....	30

## Tables

Table 2-1. 82750PD and Host CPU Access to the Memory Address Space .....	6
Table 2-2. Quad Word Interleaving for a 64-Bit SFB .....	8
Table 2-3. Dword Interleaving for a 32-Bit SFB .....	9
Table 3-1. ALU Condition Code Register .....	5
Table 3-2. ALU Operations .....	6
Table 3-3. Barrel Shifter Operations and Registers .....	8
Table 3-4. Microcode RAM Registers .....	10
Table 3-5. Core Input FIFO Registers .....	14
Table 3-6. Input FIFO Control Registers .....	14
Table 3-7. Circular Buffer Register (circbuf) .....	16
Table 3-8. Output FIFO Registers .....	19
Table 3-9. Output FIFO Control Registers .....	19
Table 3-10. Registers Associated with the Statistical Decoder .....	23
Table 3-11. Codes that Can Be Read by the Statistical Decoder .....	24
Table 3-12. Sample Code Description Tables .....	25
Table 3-13. Decoded Values .....	26
Table 3-14. Decoded Values Using END Mode .....	28
Table 3-15. END Flag Decoded Values .....	29
Table 3-16. Packed 3-bit Field Decoded Values .....	30
Table 3-17. Statistical Decoder Control Register .....	31
Table 3-18. Control Bits for the Code Description Table. ....	31
Table 3-19. SFB Bit-Stream Decode Values .....	35
Table 3-20. Decoding Symbols .....	36
Table 3-21. Pixel Interpolator Control Register .....	40
Table 3-22. Interpolator Mode Selection .....	41

## Contents

Table 3-23. Pipeline Delay .....	43
Table 3-24. Core Control Register.....	45
Table 3-25. Run, Step, and Halt Modes.....	45
Table 3-26. Core Interrupt Flag Register .....	47
Table 3-27. Core Status Register .....	48
Table 3-28. Interrupt Bits in the Flag, Control, and Status Registers.....	50
Table 3-29. Host Address Mapping of the Core Registers .....	51
Table 3-30. 82750PD A Bus Source/Destination Address .....	52
Table 3-31. 82750PD B Bus Source/Destination Address .....	53
Table 3-32. SFB Pointer Addresses.....	54
Table 3-33. Byte Offsets for the Area-c Registers.....	55
Table 4-1. PC Load Example.....	3
Table 4-2. Condition Flag Select Field Assignments.....	4
Table 4-3. Microcode Next Instruction Selection .....	4
Table 4-4. 82750PD Core Source/Destination Coding .....	6
Table 4-5. 82750PD Source/Destination Coding (Con't.) .....	7
Table 4-6. SHIFT Control Field Coding.....	10
Table 4-7. ALU Operations .....	11
Table 5-1. Host Access Modes to the Multimedia System.....	2
Table 5-2. Generic Address Configuration Fields. ....	6
Table 5-3. ISA Implementation of POS Fields .....	7
Table 5-4. EISA POS2 Register .....	7
Table 5-5. EISA POS3 Register .....	8
Table 5-6. Micro Channel POS2 Register.....	8
Table 5-7. Micro Channel POS3 Register.....	9
Table 5-8. PCI POS2 Register.....	9
Table 5-9. PCI POS3 Register.....	10
Table 5-10. VL-Bus Implementation of POS Fields.....	10
Table 5-11. The Value of IO_START .....	13
Table 5-12. Format for the Address of a Host I/O Register..	13
Table 5-13. "EMS" Configuration Registers .....	19
Table 5-14. "EMS" Page Address Registers. ....	21
Table 5-15. Indirect Address Register.....	22
Table 5-16. Indirect Data Register .....	23
Table 5-17. Host-SFB Write FIFO Registers.....	27
Table 5-18. Write FIFO Control Register.....	29
Table 5-19. Byte Select Bits in the WFCON Register .....	30



Table 5-20. I/O Sequence to Read/Write the SFB Address	34
Table 5-21. Write FIFO Test Register	39
Table 5-22. Read FIFO Registers	40
Table 5-23. Read FIFO Control Register	42
Table 5-24. Byte Select Bits in the RFCON Register	42
Table 5-25. I/O Sequence for RFCNTR Register Accesses	46
Table 5-26. Read FIFO Test Register	47
Table 5-27. Enabling SFBI Events to Generate Meta- Interrupts	51
Table 5-28. VBUS Codes	54
Table 5-29. Priorities for Generation of VBUS Codes	55
Table 5-30. General Control Register	56
Table 5-31. General Status Register	57
Table 5-32. CFG Register Numbers and Names	59
Table 5-33. CFG Register Number Register	60
Table 5-34. CFG Register Data Register	60
Table 5-35. General Configuration Register	61
Table 5-36. I/O Base CFG Register	61
Table 5-37. Bus CFG Register	62
Table 5-38. Configuration Bits for the ISA Bus	63
Table 5-39. Configuration Bits for the EISA Bus	64
Table 5-40. Configuration Bits for the PCI Bus	65
Table 5-41. Configuration Bits for the VL-Bus	66
Table 5-43. INT CFG Register	67
Table 5-44. Values of Interrupt Configuration Bits	67
Table 5-45. SFBI CFG 04 Register	68
Table 5-46. SFBI CFG 05 Register	69
Table 6-1. Examples of Signal Events	9
Table 6-2. Service Request Data Packet Definition	11
Table 6-3. Service Completion Data Packet Definition	13
Table 6-4. Other Function Code Data Packet Definition	15
Table 6-5. SynchroLink Register Map	17
Table 6-6. Transceiver Registers	18
Table 6-7. Enabling SFBI Events to Generate VBUS Codes	20
Table 6-8. XCVRn Register Modes	21
Table 6-9. XCVRn Settings for the Four Message Modes	22
Table 6-10. Priority of Transmission Requests	23
Table 6-11. Broadcast Transmission Bit Activity	25
Table 6-12. Send Service Request Bit Activity	30

Table 6-13. Receive Service Register.....	33
Table 6-14. Message Status Register.....	37
Table 6-15. Transceiver Status Bits and the SMSG Bit.....	37
Table 6-16. Match Service Register.....	38
Table 6-17. Message Configuration Register.....	40
Table 6-18. SynchroLink Clock Frequency.....	40
Table 6-19. Enabling Meta-Interrupt Events to Generate VBUS Codes .....	45

Table A-1. 82750PD Core Registers Excluded from Appendix A.....	1
Table A-2. EISA POS3 Register .....	2
Table A-3. EISA POS2 Register .....	2
Table A-4. ALU Condition Code Register .....	3
Table A-5. Core Control Register .....	4
Table A-6. I/O Base CFG Register .....	5
Table A-7. Bus CFG Register .....	5
Table A-8. CFG Register Data Register .....	6
Table A-9. General Configuration Register .....	6
Table A-10. INT CFG Register .....	7
Table A-11. CFG Register Number Register .....	8
Table A-12. SFBI CFG 04 Register.....	9
Table A-13. SFBI CFG 05 Register.....	10
Table A-14. Core Interrupt Flag Register .....	11
Table A-15. PCI POS2 Register .....	11
Table A-16. Core Status Register.....	12
Table A-17. "EMS" Configuration Registers .....	13
Table A-18. General Control Register .....	13
Table A-19. General Status Register .....	14
Table A-20. Input FIFO Control Register .....	14
Table A-21. Indirect Address Register.....	16
Table A-22. Indirect Data Register.....	16
Table A-23. Message Configuration Register .....	17
Table A-24. Micro Channel POS3 Register.....	18
Table A-25. Micro Channel POS3 Register.....	19
Table A-26. Match Service Register .....	19
Table A-27. Message Status Register .....	20
Table A-28. Core Output FIFO Control Register .....	21
Table A-29. "EMS" Page Address Registers.....	22
Table A-30. Pixel Interpolator Control Register.....	23

Table A-31. Read FIFO Address Counter Register .....	24
Table A-32. Read FIFO Control Register .....	25
Table A-33. Read FIFO Data Register .....	26
Table A-34. Read FIFO Selected Byte Register .....	27
Table A-35. Read FIFO Test Register .....	27
Table A-36. Receive Service Register .....	28
Table A-37. Statusical Decoder Control Register .....	29
Table A-38. Write FIFO Address Counter Register .....	30
Table A-39. Write FIFO Control Register .....	31
Table A-40. Write FIFO Data Register .....	32
Table A-41. Write FIFO Selected Byte Register .....	33
Table A-42. Write FIFO Test Register .....	33
Table A-43. Transceiver Registers .....	34

Table B-1. Association of 82750PD Address Bits to DRAM/VRAM Bits .....	1
Table B-2. 82750PD Internal Bus Address to DRAM/VRAM Physical Address Mapping .....	3

Contents-12

ATI017776

# Chapter 1

## Guide to this Manual

---

This manual describes the 82750PD video processor. It is written for programmers who are familiar with programming microprocessors and who have some acquaintance with video signal processing. The manual provides detailed information for writing 82750PD microcode and for programming the host processor.

### 1.1 Manual Contents

---

**Chapter 2**  
**Introduction to**  
**the 82750PD**

Provides an overview of the 82750PD and the Shared Frame Buffer (SFB) Architecture. Also included are descriptions of the memory types, register types, and the memory address space.

**Chapter 3**  
**82750PD Core**  
**Architecture**

Describes the operation of the 82750PD core, which is a re-implementation of the 82750PB processor embedded in the 82750PD.

**Chapter 4**  
**Microcode**  
**Instruction**  
**Format**

Outlines the fields of the microcode instructions and describes the instruction sequencing.

**Chapter 5  
Universal Host Bus  
Interface**

Describes the 82750PD Universal Host Bus Interface (UHBI) for the host CPU. The UHBI can accommodate the following bus types: ISA, EISA, Micro Channel\*, PCI, and VL-Bus.

**Chapter 6  
SynchroLink\***

Describes the SynchroLink communications bus, which supports the 82750PD communications with other devices on the Shared Frame Buffer Interconnect (SFEI).

**Appendix A  
82750PD  
Registers**

Provides detailed descriptions of the 82750PD registers.

**Appendix B  
Logical and Device  
Addresses**

Specifies the correspondence between 82750PD logical address bits and VRAM/DRAM physical address bits.

**Appendix C  
Programming  
Examples**

Shows illustrative examples of programming the 82750PD in assembly language and "C" language.

---

\* All products mentioned are trademarks of their respective companies.

## 1.2 Notational Conventions and Terminology

<i>italics</i>	Italics are used to introduce terminology. Italics are also used for 82750PD core register names and for some variables. See "Register Names" and "Variables" in this section.
Register Names	Mnemonics for registers inside the 82750PD core are written in italics. For example, the mnemonic for the Field Counter Register is <i>fcnrl</i> . Mnemonics for 82750PD registers outside the 82750PD core are written in upper case. The mnemonic for the General Control Register is GENCON.
Variables	Variables (indices) in register mnemonics are shown in upper case or lower case and in italics or non-italics to contrast with the mnemonic.  For example, the three "EMS" Configuration Registers have the mnemonics EMSCFG <i>n</i> ; <i>n</i> = 1-3. The individual registers are EMSCFG1, EMSCFG2, and EMSCFG3.  The two Input FIFO Control Registers in the 82750PD core have the mnemonics <i>in</i> N- <i>c</i> ; <i>N</i> = 1,2. Individually, they are written as <i>in</i> 1- <i>c</i> and <i>in</i> 2- <i>c</i> .
Reserved. RSVD	Certain register bytes and bits are labeled "Reserved" or "RSVD." A reserved register byte or bit is nonfunctional in the current product but may have a function in a later product. Unless otherwise noted, reading a reserved bit or byte returns an unspecified value. For compatibility with future products, write a reserved byte as 00h and a reserved bit as '0'.
Illegal	A bit combination marked "illegal" should not be used. If it is used, the results are unspecified. The following bit table shows an example:

Bit 1	Bit 2	
0	0	Single point mode
0	1	Random mode
1	0	Sequential mode
1	1	Illegal

The bit combination '1 1' should not be used. If it is used, the operation is unspecified.

**Set and Clear** The terms *set* and *clear* describe the state of a bit or the act of driving a bit to that state. A bit with the value '0' is *clear*. To *clear* a bit means to drive it to '0'. A bit with the value '1' is *set*. To *set* a bit means to drive it to '1'.

**Numbers** Hexadecimal numbers are represented by a string of digits followed by the character *h*. If the number would otherwise begin with the letter A, B, ..., or F, a prefix '0' is added. For example, FA6h is written as 0FA6h.

Binary numbers are written with an appended "b" in cases where the base may not be clear. The binary value six could be written as "110" or "110b".

**Abbreviations** LSB = least significant bit  
MSB = most significant bit

**Units of Measure** The following symbols are used to represent units of measure:

Kbyte kilobyte = 1024 bytes  
Mbyte megabyte = 1024 kilobytes  
Gbyte gigabyte = 1024 megabytes  
MHz megahertz  
 $\mu$ s microseconds

**Related Documents** i750™, i860™, and i96™ Processors and Related Products: Intel Corporation, 1993 (Order Number: 272084-002).



82750PD Video Processor. Intel Corporation. 1993.  
(Order Number: 272341-001). 1750 Pixel Processor  
Microcode Development Tools User's Guide. Intel  
Corporation. 1993 (Order Number: 485407-001)

AT1017781

## Chapter 2

# Introduction to the 82750PD

---

### 2.1 Introduction

---

The 82750PD is a programmable video processor that supports a variety of video compression algorithms. The core of the 82750PD is its predecessor, the 82750PB, which includes a variable length sequence decoder and a pixel interpolator. The 82750PD is microcode-compatible with the 82750PB.

The 82750PD's design is based on a Shared Frame Buffer (SFB) architecture. The Shared Frame Buffer Interconnect (SFBI) is a multi-master bus that interconnects the SFB, the 82750PD, a graphics processor, and other multimedia system components, as shown in Figure 2-1. This architecture provides shared access to the SFB (video memory) and results in an integrated video and graphics subsystem.

Figure 2-2 is a closer look at the 82750PD itself. The Universal Host Bus Interface (UHBI) supports several bus types: ISA, EISA, Micro Channel, PCI, and VL-Bus. The bus type is selected by strapping inputs. The UHBI has a pair of Host-SFB FIFOs, which provide the host with high-speed access to the Shared Frame Buffer via the internal bus.

Devices on the SFBI communicate via the SynchroLink, a serial bus that provides for synchronization of graphics, video, and audio events without using interrupts to the host.

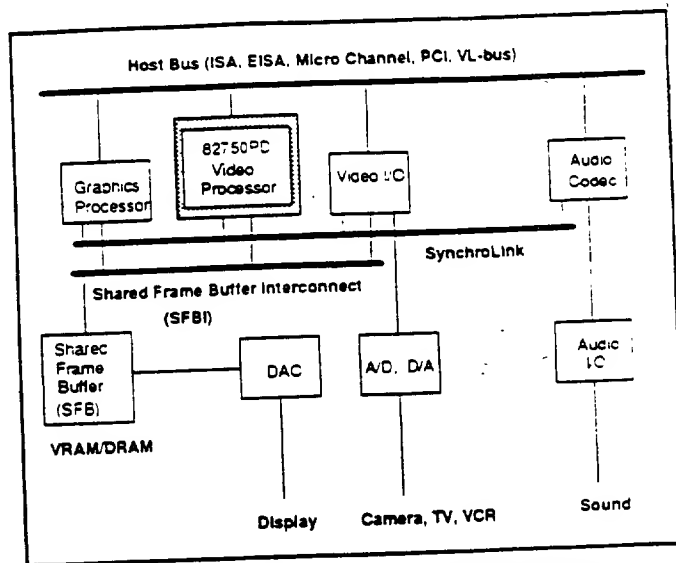


Figure 2-1. Shared Frame Buffer System

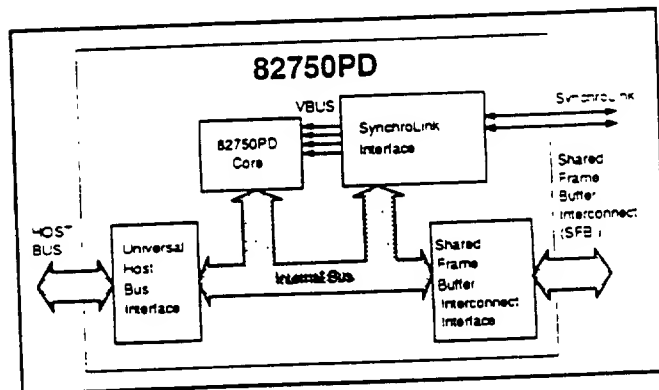


Figure 2-2. 82750PD Block Diagram

## 2.2 Memory and Registers

Figure 2-3 is a 82750PD block diagram showing the on-chip memory and registers that are accessible by the 82750PD core and/or the host computer.

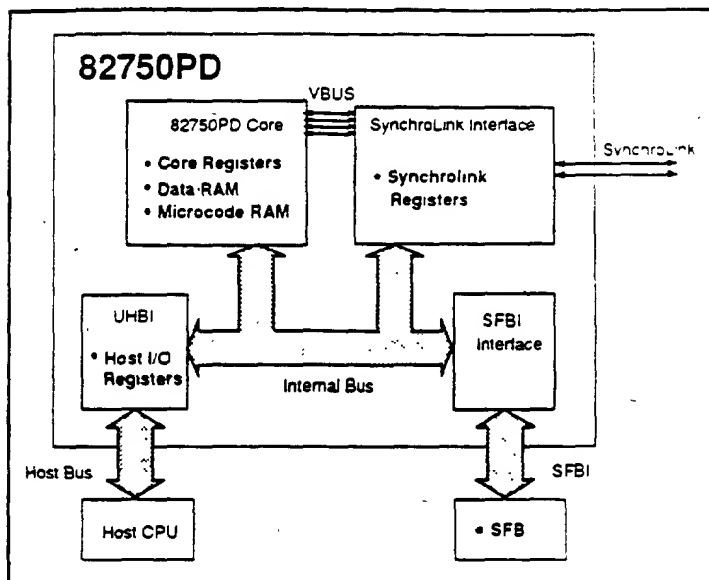


Figure 2-3. Block Diagram Showing Memory and Registers

The 82750PD's memory and registers include these categories:

- Data RAM and Microcode RAM.
- Core Registers. These registers reside in the core. The host can access the core registers via the 82750PD internal bus (the on-chip address/data bus, which is external to the core). The 82750PD can access these registers

only through microcode instructions. The core registers are described in Chapter 3.

- **SynchroLink Registers.** These registers are used by the 82750PD in its communications with other devices on the SFB. They are accessible from the 82750PD and the host via the internal bus. These registers include:
  - twelve transceiver registers.
  - four registers for configuration, status, and control.

Chapter 6 describes the event synchronization registers.

- **Host I/O Registers.** This group includes registers for the Universal Host Bus Interface and registers for the SFB interface. The 82750PD cannot access these registers. They are accessible from the host either directly in its I/O space or indirectly through its I/O registers. The following register categories comprise the host I/O registers.
  - "EMS" page address registers.
  - Host-SFB FIFO registers
  - Indirect I/O registers
  - Configuration Registers
  - General status and control registers

The host I/O registers are described in Chapter 5.

## 2.3 82750PD Memory Address Space

The addresses of the following memory and register groups are in the 82750PD memory address space:

- the Shared Frame Buffer (SFB)
- the core registers

- the SynchroLink registers

(Note that the host I/O registers, which are associated with the Universal Host Bus Interface, are accessed in the host computer's I/O space and are not accessible from the core.)

The memory address space can be accessed by the host as well as the 82750PD. However, not all registers and memory are accessible from both the 82750PD and the host. Further, the host can access this address space by high-speed Host-SFB FIFOs (see Chapter 5) and also by using the indirect I/O and "EMS" modes (which access a common set of memory and registers).

Table 2-1 shows the memory address space as it is accessed by:

- the 82750PD core.
- the host via the Host-SFB FIFOs.
- the host via "EMS" and indirect I/O accesses.

Table 2-1. 82750PD and Host CPU Access to the Memory Address Space

Address	Accessible from the 82750PD	Accessible from the Host PC via Host-SFB FIFOs	Accessible from the Host via "EMS" and indirect I/O Modes
FFFFFFFFh	RESERVED	Not Accessible	Not Accessible
01800000h 017FFFFFFh	SFB Interleaved* ODD Quad Words	Not Accessible	Not Accessible
01400000h 013FFFFFFh	SFB Interleaved* EVEN Quad Words	Not Accessible	Not Accessible
01000000h 00FFFFFFh	SynchroLink Registers	SynchroLink Registers	SynchroLink Registers
00FF0000h	Access byte, word, dword, dword aligned	Access byte, word, dword, dword aligned	Access byte, word, dword, dword aligned
00FEFFFFh	RESERVED	RESERVED	Core Registers
00FE0000h 00FDFFFFh	RESERVED	RESERVED	Access word, word aligned
00800000h 007FFFFFFh	SFB	SFB	SFB
00000000h	Access byte, word, nonaligned	Access byte, word, nonaligned	Access byte, word, dword, nonaligned

\* An example of interleaving is given in Section 2.5

## 2.4 82750PD Core Registers

The 82750PD core registers (i.e., the registers located inside the 82750PD core) can be accessed by both the 82750PD and the host. The 82750PB core accesses these registers in microcode; it cannot access them as memory locations in the memory address space. The host, using the indirect I/O or "EMS" modes, accesses the core registers at addresses 00FE0000h—00FEFFFFh in the 82750PD memory address space (see Table 2-1).

## 2.5 Shared Frame Buffer (SFB)

The Shared Frame Buffer (SFB) is memory connected to the Shared Frame Buffer Interconnect (SFB I). For a 64-bit SFB this area of memory comprises 8 Mbytes in the range 00000000h—007FFFFFFh. As noted in the following subsections, the 82750PD can also access this memory at locations in the range 01000000h—017FFFFFFh. A 32-bit SFB occupies 4 Mbytes in the range 00000000h—003FFFFFFh, and can also be accessed at locations in the range 01000000h—013FFFFFFh.

### 2.5.1 Accessing the 64-Bit SFB

The host can access the 64 bit Shared Frame Buffer via the Host-SFB FIFOs, the indirect I/O mode, and the "EMS" mode. The 82750PD can access each quad word in the 8-Mbyte SFB at two addresses:

- Consecutive **even** quad words in the address range 00000000h—007FFFFFFh can also be accessed in the 4-Mbyte range: 01000000h—013FFFFFFh.



- Consecutive **odd** quad words in the range 00000000h—007FFFFFFh can also be accessed in the 4-Mbyte range: 01400000h—017FFFFFFh.

Table 2-3 shows how the first few quad words of the 64-bit SFB are mapped into the even quad word and odd quad word areas. The columns in the center show the quad words in lower memory with the bytes numbered. The column on the left shows the unmapped quad word addresses. The column on the right lists the mapped quad word addresses.

Table 2-2. Quad Word Interleaving for a 64-Bit SFB

Quad Word Addresses (Unmapped)	Numbered Bytes								Mapped Quad Word Addresses
0000 0000h	7h	6h	5h	4h	3h	2h	1h	0h	0100 0000h
0000 0008h	Fh	Eh	Dh	Ch	Bh	Ah	9h	8h	0140 0000h
0000 0010h	17h	16h	15h	14h	13h	12h	11h	10h	0100 0008h
0000 0018h	1Fh	1Eh	1Dh	1Ch	1Bh	1Ah	19h	18h	0140 0008h
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

### 2.5.2 Accessing the 32 Bit SFB

The host can access the the 32-bit Shared Frame Buffer via the Host-SFB FIFOs, the indirect I/O mode, and the "EMS" mode. The 82750PD can access each dword in the 4-Mbyte SFB at two addresses:

- Consecutive **even** dwords in the address range 00000000h—003FFFFFFh can also be accessed in the 2-Mbyte range: 01000000h—011FFFFFFh.

- Consecutive **odd** dwords in the range 00000000h—003FFFFFh can also be accessed in the 2-Mbyte range: 01200000h—013FFFFFh.

Table 2-3 shows how the first few dwords of the 32 bit SFB are mapped into the even dword and odd dword areas. The columns in the center show the dwords in lower memory with the bytes numbered. The column on the left shows the unmapped dword addresses. The column on the right lists the mapped dword addresses.

Table 2-3. Dword Interleaving for a 32-Bit SFB

Dword Addresses (Unmapped)	Numbered Bytes				Mapped Dword Addresses
0000 0000h	3h	2h	1h	0h	0100 0000h
0000 0004h	7h	6h	5h	4h	0140 0000h
0000 0008h	Bh	Ah	9h	8h	0100 0004h
0000 000Ch	Fh	Ek	Dh	Ch	0140 0004h
.	.	.	.	.	.
.	.	.	.	.	.

### 2.5.3 Types of Memory on the SFB

All memory connected to the SFB Interconnect is considered a part of the Shared Frame Buffer. The SFB can be subdivided into "display memory" (memory for storing the screen pixels) and "off-screen memory." The off-screen memory is subdivided into "82750PD workspace," "graphics workspace" (for the graphics processor), and "other memory." While the display memory must be implemented in VRAM, the 82750PD workspace and the graphics workspace can be implemented in VRAM or DRAM.

## 2.6 Reserved Memory Locations

---

Memory locations marked as "Reserved" or "RSVD" should not be accessed. To prevent the 82750PD and host interface from hanging, do not write software to access reserved locations; future products may define these regions to serve new functions. See table 2-1 for reserved addresses

## Chapter 3

# 82750PD Core Architecture

---

The 82750PD core is a re-implementation of the 82750PB architecture with the necessary changes to embed it in the 82750PD. This chapter describes the 82750PD core architecture and discusses the operation of the core FIFOs, the Statistical Encoder, and the Pixel Interpolator.

### 3.1 Overview

---

The 82750PD core includes a wide instruction word processor that comprises a number of processing, storage, and input/output elements. The wide instruction word architecture allows a number of these elements to operate in parallel. The various elements are connected via two 16-bit buses, the A bus and B bus, as shown in Figure 3-1. During each instruction execution cycle, data can be transferred from a bus source to a bus destination on the A and B buses. The 82750PD core executes one instruction every internal clock. The internal clock runs at one-half the frequency of the SFBI clock. The 82750PD core data and address lines are connected to the 82750PD internal bus, which is inside the 82750PD but external to the 82750PD core. Signals from the SynchroLink interface enter the core on the VBUS.

The core accesses the core registers only in microcode (see Chapter 4). The host accesses the core registers in the 82750PD memory address space. The core register addresses are given in Section 3.15.

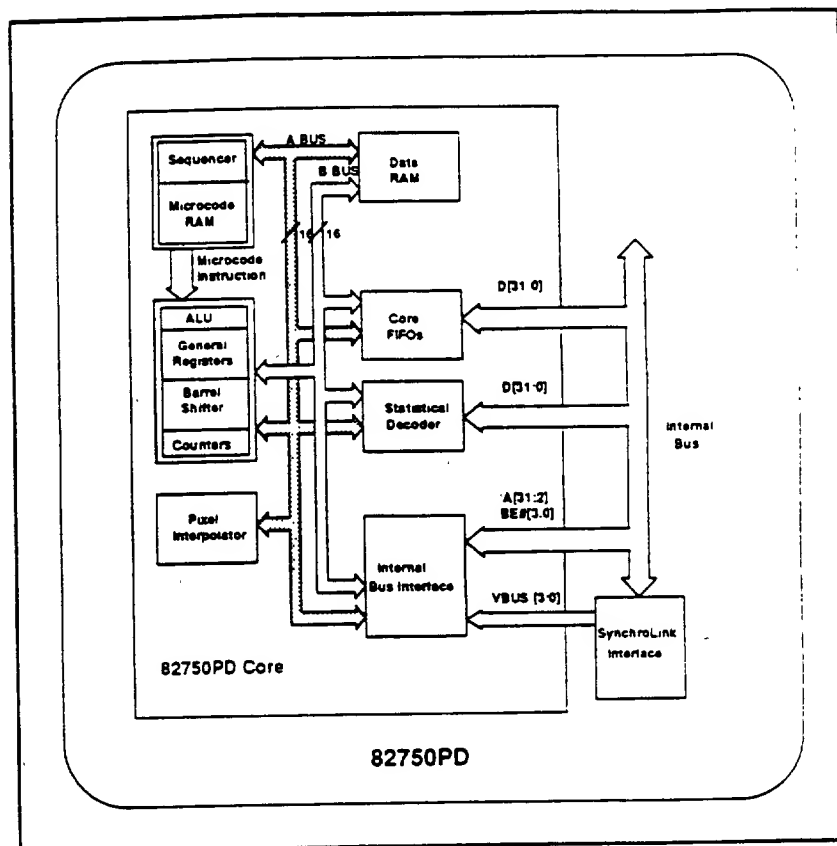


Figure 3-1. 82750PD Core Block Diagram

### 3.2 General Purpose Registers

The 82750PD core has 16 general purpose data registers, each 16 bits wide, which are connected to both the A bus and B bus as both sources and destinations. These registers, denoted by *r0-r15*<sup>1</sup>, constitute the General Registers in Figures 3-1 and 3-2. All of the registers are functionally identical except *r0*, which is described in the following paragraph. A register can be the source for both the A bus and B bus in the same instruction. However, a register cannot be the destination for both the A bus and the B bus in a single instruction. Because the registers are doubly latched, the same register can be both a source and destination in the same cycle. The result is that the data in the register prior to the current cycle is driven on the source bus, and the data on the destination bus is latched into the register at the end of the clock cycle.

The *r0* register has additional logic to allow bit shifting and byte swapping. In addition, the MSB and LSB of the *r0* register are stored in the ALU Condition Flag Register, *cc*, described in the following section. The value in *r0* can be shifted left or right one bit position per instruction cycle. For right shifts, the new MSB is equal to the old MSB; i.e., the value is sign-extended. For left shifts, the new LSB is equal to '0'. Register *r0* cannot be shifted and loaded in the same instruction.

<sup>1</sup> 82750PD core registers are shown in lower case italics. (See Section 1.2 for register notation.)

Byte swapping only occurs when *r0* is being loaded with a value from the A bus or B bus. Byte swapping causes the most significant byte and the least significant byte of the 16-bit value being loaded into *r0* to be interchanged. Refer to Chapter 4 for a description of the SHFT microcode field that controls the shifting and swapping operation in *r0*.

### 3.3 Arithmetic/Logic Unit (ALU)

The ALU performs 16-bit arithmetic and logic operations, and can also be operated as two independent 8-bit ALU's for the Dual-Add-with-Saturate operation (discussed later in this section). Two fields in the microcode instruction affect the operation of the ALU. The ALUOP field specifies the operation to be performed, and the ALUSS field specifies the source of the two ALU inputs. Refer to Chapter 4 for further information on these fields.

The two ALU operands come from values held in the ALU input latches or from "eavesdropping" on the A or B bus. The result of any ALU operation is latched in the 16-bit ALU Output Register, *alu*. In a subsequent instruction, this result can be transferred to any A or B bus destination.

#### 3.3.1 Condition Code Register

The ALU has four condition flag outputs — ALU Carry Out, ALU Sign, ALU OVerflow, and ALU Zero — which are stored in the Condition Code Register, *cc*. Table 3-1 shows the *cc* register and defines the ALU condition code outputs. For most ALU operations the states of these four condition flags are latched when the operation is complete. The exceptions are given in Section 3.3.2.

Table 3-1. ALU Condition Code Register

Mnemonic: *cc*

Address: Core Register\*

Access: See bit descriptions.

Reset State: Not available.

Bit No.	15-8	7	6	5	4	3	2	1	0
Name	RSVD	<i>r0</i> MSB	<i>r0</i> LSB	LCNTZ	ALU Zero	ALU Sign	ALU Overflow	ALU Carry Out	0

\* See Section 3.15 for core register addresses

Bit No.	Name	Description
0	RSVD	
1	ALU Carry Out	(R/W) The value of this bit equals the carry out of the most significant bit position in the ALU. This bit is '0' for all logical operations.
2	ALU Overflow	(R/W) The value of this bit is (ALU Carry Out) XOR (Carry in to most significant bit of ALU result). This bit is '0' for all logical operations.
3	ALU Sign	(R/W) The value of this bit is the most significant bit of the ALU result.
4	ALU Zero	(R/W) This bit is set only if all bits of the ALU result are '0'.
5	LCNTZ	(R/O) This bit is set only if the value of the loop counter is '0'.
6	<i>r0</i> LSB	(R/O) The value of this bit is the value of the least significant bit of register <i>r0</i> .
7	<i>r0</i> MSB	(R/O) The value of this bit is the value of the most significant bit of register <i>r0</i> .
15-8	RSVD	

### 3.3.2 ALU Operations

Table 3-2 is a list of the ALU operations. The condition flags in the *cc* register are latched upon completion of the ALU operations, with the exceptions of the eight operations tagged with asterisks in Table 3-2. These operations do not disturb the condition flags of the previous ALU operation. Microcode routines can read and write the *cc* register to save and restore the states of these flags.



Table 3-2. ALU Operations

Operation	Operation
No Operation*	Increment a
Pass a	Increment b
Pass b	Decrement a
1's Complement of a	Decrement b
1's Complement of b	Dual Add with Saturate*
a AND b	a - b - (Previous Carry
(NOT a) AND b	a - b - (Previous Borrow
a AND (NOT b)	-a - b - (Previous Borrow
a OR b	Interrupt Host*
a XOR b	Zero*
a + b	Pass a Don't Latch Flags*
a + b - 1	Pass b Don't Latch Flags*
a - b	(NOT a) OR b
-a + b	a OR (NOT b)
2's Complement of a	Dual Subtract with Saturate*
2's Complement of b	Performance Monitor*

The Dual-Add-with-Saturate operation performs independent, 8-bit ADDs on the upper and lower bytes of the two ALU operands. The two bytes of the A operand are treated as unsigned binary numbers (00:FFh corresponds to 0:255). The two bytes of the B operand are treated as offset binary numbers with an offset of +128 (00:FFh corresponds to -128:127). The upper and lower byte results, including the carry output of each byte, are treated as 9-bit offset binary numbers with a +128 offset (000:1FFh corresponds to -128:383) and are saturated to a range of 0-255. A result less than zero is set equal to '0' (00h), and a result greater than +255 is set equal to +255 (FFh).

In fact, the Dual-Add-with-Saturate operation is symmetric with respect to the two operands. Either the A operand or the B operand can be defined as

the unsigned binary value, and the other operand is treated as the offset, signed binary value.

Dual-Subtract-with-Saturate is similar to Dual-Add-with-Saturate. It calculates  $A - B + 128$  on each 8-bit half of the two 16-bit inputs, and clamps the results to 0 and 255. This can be viewed as subtracting an offset-binary signed byte (-128 to 127) from an unsigned byte (0 to 255).

### **3.3.3 82750PD Interrupts**

---

The "interrupt host" operation produces the MCINT (microcode interrupt) condition, which can potentially generate an 82750PD interrupt. This is one of several 82750PD interrupt sources, which are discussed in Sections 3.14 and 5.7.

### **3.3.4 Performance Monitoring**

---

The "performance monitor" operation toggles the PMON# pin, and is primarily used for performance monitoring and/or debugging. The PMON# function is available only when the ISA or PCI host bus is selected.

## **3.4 Barrel Shifter**

---

The barrel shifter performs a single-cycle, n-bit shift, left or right. It operates independently of the ALU and affects none of the condition flags. Table 3-3 describes the three types of shifts that can be performed. Each type of shift has an associated register, which is named for the shift operation. The A bus registers *shift-r*, *shift-rl*, and *shift-l* specify the type and length of a shift.

Table 3-3. Barrel Shifter Operations and Registers

Shift Operation (and Register Name)	Description
<i>shift-r</i>	Right shift with sign extension
<i>shift-fl</i>	Right shift with '0' fill
<i>shift-l</i>	Left shift with '0' fill
<i>shift</i>	Stores the result of the shift

To invoke a shift operation, write the 4-bit shift amount to the register named for the desired shift. The operand is taken from the B bus, and the result is stored in the Shifter Result Register, *shift*. As in the case of the ALU Result Register, the value in *shift* can be read onto the A bus or B bus in the instruction cycle that follows.

### 3.5 Data RAM

The Data RAM (DRAM) holds 512, 16-bit words, which are accessed via four pointers: *dramN*,  $N = 1-4$ . To access a specific location in DRAM, the microcode routine loads a pointer with the address to be accessed and then performs a read or write via the same pointer. The pointer can optionally be post-incremented or post-decremented in parallel with the DRAM access.

The four pointers, *dram1-dram4*, can be written and read via the A bus. When a DRAM pointer, which is only 9 bits wide, is read onto the A bus, the upper seven bits of the A bus are cleared.

**NOTE**

*The width of the DRAM pointers may change in later versions of the 82750PD. Software should not rely on the width of a pointer to, for example, mask the upper seven bits of a value to '0'.*

All four pointers can be used to read or write the Data RAM from either the A bus or B bus. Only one DRAM access can be performed in a single clock cycle.

The notation for a DRAM access is (using C-like language syntax) `*dram1`. The `*` means "the value pointed to by". As another example, `*dram3++` means access the Data RAM using the pointer `dram3` and then increment `dram3`. The symbol `--` in place of the `++` indicates autodecrement.

### 3.6 Loop Counters

Microcode programs can use two 16-bit loop counters, `cnt1` and `cnt2`, for automatically counting iterations of a microcode loop. In parallel with other operations performed in an instruction, the 82750PD core can decrement either loop counter, and execute a conditional branch based on the loop counter value being equal or not equal to '0'.

Refer to Section 4.3 for descriptions of the following microcode bits and flags associated with the loop counters:

- LC, the Loop Counter Select bit
- CNT, the Decrement Loop Counter bit
- LCNTZ, the Loop Counter Zero value in the Condition Flag Select (CFSEL) field of the microcode instruction

The two loop counters (*cnt1* and *cnt2*) can be read or written on the A bus and, therefore, can be used for variable storage when they not serving as loop counters. You can also write to and decrement a loop counter in the same instruction cycle by specifying the counter as A DEST and setting the CNT bit. The value of the counter at the start of the next cycle is the newly loaded value of the counter minus one. Note that the LC microcode bit does not affect the loop counter that is written or read over the A bus, because each loop counter is separately addressable as an A bus source or destination.

### 3.7 Microcode RAM

The 82750PD executes instructions stored in Microcode RAM (MRAM), which is inside the core. The MRAM holds 512 48-bit instructions. To start the microcode processor, the host CPU normally loads a microcode program into the MRAM, points the program counter, *pc*, to the start of the program, and then clears the HALT bit in the *ccontrol* register. The microcode processor can also load its own MRAM to overlay new routines. Table 3-4 lists the registers associated with the microcode RAM.

Table 3-4. Microcode RAM Registers

Register Mnemonic	Register Name	Description
<i>mcode1</i> – <i>mcode3</i>	Microcode Instruction Register 1–3	<i>mcode1</i> – <i>mcode3</i> are loaded with the three words of a 48-bit instruction to be written to MRAM
<i>maddr</i>	Microcode Address Register	A pointer to locations in microcode RAM
<i>pc</i>	Program Counter	A pointer to the next microcode instruction to be executed

The host CPU can write an instruction to MRAM by executing the following steps:

1. Load the three registers *mcode1-mcode3* with the three 16-bit words of the instruction (the most significant word goes into *mcode1*). The order of loading does not matter.
2. Load *maddr* with the address where the instruction is to be written. This initiates the write to microcode RAM.

The host CPU can read the MRAM with this sequence:

1. Load the *pc* with the address of the instruction to be read.
2. Read the three 16-bit words of the instruction from the *mcode1-mcode3* registers.

Normally, this would be done by the host CPU while the 82750PD is halted.

The *mcode1-mcode3* registers used in a read instruction and the write *mcode1-mcode3* registers used in a write instruction are, in fact, different registers. Writing values into *mcode1-mcode3* and then reading the values of *mcode1-mcode3* does not return the same values as just written. The read registers hold the instructions stored in the instruction latch (the instruction to be executed). The write registers hold an instruction that is about to be written into microcode RAM.

After writing to *maddr* to load an instruction into microcode RAM, a one-cycle freeze occurs. During the freeze, the instruction is written to microcode RAM. The instruction following the write to *maddr* can either jump to the address just loaded or start loading the *mcode1-mcode3* registers with the next instruction to be written.

The 82750PD requires at least one instruction between the write to *maddr* and the execution of the instruction that is loaded by the write to *maddr*. See Section C.3 for an illustrative example of code.

When the host CPU writes a microcode-RAM address to the *pc*, the instruction at that address is loaded into the *mcode1-mcode3* registers. When the microcode processor is released from its Halt condition, this is the first instruction executed.

When the host CPU reads the *pc*, the result returned is the address of the instruction that is executed when Halt is released, that is, the address of the instruction held in the *mcode1-mcode3* registers.

### 3.8 Horizontal Line Counter

---

The 12-bit Horizontal Line Counter, *hcnt*, is updated by VBUS codes. (See Section 5.7 for the generation of VBUS codes.) The counter is cleared by a VODD code and incremented by an HLINE code. A value can also be written into the Horizontal Line Counter by microcode or the host. The upper four bits of *hcnt* always read '0's.

### 3.9 Field Counter

---

The 4-bit Field Counter, *fcnt*, is updated by VBUS codes. The counter is incremented each time a VODD code or VEVEN code is received. When the field counter is read, the upper 12 bits read '0's. This counter is not initialized by reset.

### 3.10 Core Input FIFOs

The two Core Input FIFOs are a pair of high-speed input channels that read data from the SFB via the internal bus. These FIFOs are distinct from the 82750PD host-SFB FIFOs, which reside outside the 82750PD core and transfer data between the SFB and the host (see Section 5.6).

(The Core Input FIFOs can read from any address on the internal bus that is accessible from the 82750PD — not just the SFBI. However, the description of the FIFO operations is simplified by assuming that the FIFO is reading the SFB.)

The two input FIFOs are independent. They can read the SFB in bytes, beginning with any byte, or in words, beginning on any word boundary. To speed the data input process, you can program the FIFO to automatically increment or decrement the SFB address and to increment/decrement by words or dwords. To obtain the requested bytes/words, the FIFO fetches the data in quad words (64 bits), which are then read by the microcode. The FIFO's double buffered design enables it to fetch the next quad word while the microcode reads the bytes/words from the current quad word in the FIFO. (This quad-word fetching operation is transparent to the programmer.)

A FIFO is denoted by FIFO $n$ , where  $n = 1$  or  $2$ . Table 3-5 lists the registers associated with the Core Input FIFOs.



Table 3-5. Core Input FIFO Registers

Register Mnemonic	Register Name	Description
<i>in1-lo, in2-lo</i>	Core Input FIFO <sub>n</sub> Low Address	Specifies bits 15–0 of the address in 82750PD memory address space
<i>in1-hi, in2-hi</i>	Core Input FIFO <sub>n</sub> High Address	Specifies bits 31–16 of the address in 82750PD memory address space
<i>in1-c, in2-c</i>	Core Input FIFO <sub>n</sub> Control	Each of these registers controls its corresponding FIFO.
<i>*in1, *in2</i>	Core Input FIFO <sub>n</sub> Data	Reading this register reads the (byte/word) fetched over the internal bus by the Core Input FIFO
<i>arcbuf</i>	Circular Buffer	Specifies the size of the circular buffer and forces the associated address bit low

InN-lo, InN-hi Table 3-6 shows the form of the control registers, in1-c, in2-c. The paragraphs to follow describe the register bits.

Table 3-6. Input FIFO Control Registers

Mnemonic: *inN-c*; N=1,2

Address: Core Register\*

Access: R/W

Reset State: Not available.

Bit No.	15–6	5	4	3	2	1	0
Name	RSVD	BY-32 MODE	CB	PF-OFF	AHOLD	DEC/ INC#	BYTE WORD#

\* See Section 3.15 for core register addresses

**WORD/BYTE** (bit 0) (See also the BY-32 Mode bit.).

1 = Byte mode. The FIFO reads bytes. The high byte of *\*inN* holds the value 00h.

0 = Word mode. The FIFO reads words beginning on any word boundary.

**DEC/INC#** (Decrement/Increment#, bit 1). This bit determines the order of reading bytes/words from the SFB, as shown below:

1 = Decrement mode. The FIFO reads from the most significant to the least significant byte or word.

0 = Increment mode. The FIFO reads from the least significant to the most significant byte or word.

**AHOLD** (Address Hold, bit 2) Setting this bit disables automatic incrementing/decrementing of *irN-lo* and *irN-hi*, and prevents the Core Input FIFOs from double buffering the read data. At the end of an internal bus cycle the FIFO is updated with 64 bits of data. The Core Input FIFO does not issue another read request to the internal bus until there is a write to *irN-lo* or a roll-over/roll-under read access of the Core Input FIFO. If there is a write to *irN-lo*, the FIFO then fetches data from the new location. If a roll-over/roll-under occurs, a memory request is issued to fetch data from the unchanged address.

**PF-OFF** (Prefetch Off, bit 3). Setting this bit causes the Core Input FIFO to wait for a request to fetch a new quad word over the internal bus.

1 = Prefetch-Off mode. The FIFO prefetches the first two quad words to fill its buffer (when started at a new address location) but thereafter fetches a new quad word only when there is a read request for a byte/word in an unfetched quad word.

0 = Prefetch-On mode. The Core Input FIFO prefetches successive quad words as necessary to keep its buffer full. Fetch addresses ascend or descend according to the INC/DEC bit.

**CB** (Circular Buffer, bit 4). Setting this bit enables the creation of a circular buffer in the SFB. The appropriate address bit on the internal bus (depending on the size of the circular buffer to be created) is cleared. **The register pointers remain**

**unchanged.** The size of the circular buffer can be 64 Kbytes, 128 Kbytes, or 256 Kbytes, as determined by bits 2-0 of the Circular Buffer Register, *crcbuf* (shown in Table 3-7).

Table 3-7. Circular Buffer Register (*crcbuf*)

Bits 2-0	Buffer Size	Effect on 82750PD Internal Address Bus (For Function Enabled)
000	Disabled	None
100	256 Kbytes	Internal Address 18 forced to 0
010	128 Kbytes	Internal Address 17 forced to 0
001	64 Kbytes	Internal Address 16 forced to 0

**BY-32 MODE** (bit 5). This bit is a "don't care" if prefetches are disabled (PF-OFF = 1).

1 = The SFB pointer increments or decrements by four bytes.

0 = The SFB pointer increments or decrements by two bytes.

The WORD/BYTE bit and the PF-OFF bit determine which bytes or words are read when the SFB address is automatically incremented/decremented.

Figure 3-2 shows the data words obtained for the four combinations of the WORD/BYTE and BY-32 MODE bits.

WORD/BYTE = 0 BY-32 MODE = 0	WORD/BYTE = 0 BY-32 MODE = 1
Begin at byte 1.	Begin at byte 1.
"00h"   BYTE 1	"00h"   BYTE 1
"00h"   BYTE 3	"00h"   BYTE 5
"00h"   BYTE 5	"00h"   BYTE 9
WORD/BYTE = 1 BY-32 MODE = 0	WORD/BYTE = 1 BY-32 MODE = 1
Begin at byte 0.	Begin at byte 0.
BYTE 1   BYTE 0	BYTE 1   BYTE 0
BYTE 3   BYTE 2	BYTE 5   BYTE 4
BYTE 5   BYTE 4	BYTE 9   BYTE 8

Figure 3-2. Input FIFO Modes for Reading Data

The following steps are the standard sequence for initializing a Core Input FIFO and beginning a read operation:

1. Write to the control register (*inN-c*).
2. Using a single instruction, write to the high address register (*inN-hi*) and low address register (*inN-lo*). (After this write the FIFO begins fetching bytes or words from the SFB via the internal bus.)
3. Read a byte or word from *\*inN*.

Successive reads from *\*inN* read sequential bytes or words from the SFB. Writing to the control register each time the FIFO is started at a new address is unnecessary, except to change the FIFO mode. Further, if the new address is within the same 64-Kbyte page of the SFB, only the lo-address need

be written to start the FIFO reading from the new address. Any old data in the FIFO is lost.

If microcode attempts to read a value from an empty Core Input FIFO, the processor is frozen prior to the execution of the next instruction. It remains frozen until the FIFO control logic has fetched another dword over the internal bus and extracted the next value. At this point the processor is released, and the instruction that reads the value is executed. When the processor is frozen and waiting for an empty FIFO, that FIFO's SFB access priority is raised above all other FIFO's.

### 3.11 Core Output FIFOs

---

The Core Output FIFOs, are high-speed output channels for writing data over the internal bus to the SFB. These FIFOs are distinct from the 82750PD's host-SFB FIFOs, which are external to the 82750PD core and transfer data between the SFB and the host (see Section 5.6).

The Core Output FIFOs operate independently, each with its own set of registers. The FIFOs can write to any address on the internal bus that is accessible to the 82750PD. To simplify the discussion, we assume that the FIFOs are writing to the SFB.

Data can be written to the SFB in bytes/words beginning on any byte/word boundary, respectively. To speed the data input process, you can program the FIFO to automatically increment or decrement the SFB address by words or dwords. The FIFO collects the bytes/words written to it and writes them to the SFB in quad words. The FIFO's double buffered design permits the microcode to write the bytes/words of one quad word while the FIFO writes the previous quad word to the SFB. (The actual

process of assembling the quad words and writing to the SFB is transparent to the programmer.)

Table 3-8 describes briefly the registers associated with the Core Output FIFOs.

Table 3-8. Output FIFO Registers

Register Mnemonic	Register Name	Description
<i>out1-lo, out2-lo</i>	Output FIFO $n$ Low Address	Specifies bits 15–0 of the address in the 82750PD memory address space
<i>out1-hi, out2-hi</i>	Output FIFO $n$ High Address	Specifies bits 31–16 of the address in the 82750PD memory address space
<i>out1-c, out2-c</i>	Output FIFO $n$ Control	Each of these registers controls its associated FIFO
<i>*out1, *out2</i>	Output FIFO $n$ Data	A byte/word written to this register is written to the internal bus by the FIFO
<i>out1+*, out2+*</i>	Output FIFO $n$ Skip Data	A byte/word written to this register is skipped by the FIFO. It is not written to the 82750PD address space.
<i>circbuf</i>	Circular Buffer	Specifies the size of the circular buffer and forces the associated address pin low

Each Core Output FIFO has its own control register, *out1-c* or *out2-c*, which is described in Table 3-9 and the following bit descriptions..

Table 3-9. Output FIFO Control Registers

Mnemonic: *outN-c*; N=1,2

Address: Core Register\*

Access: R/W

Reset State: Not available.

Bit No.	15-6	5	4	3	2	1	0
Name	RSVD	BY-32 MODE	FORCE-LSB ENABLE	FORCE LSB VALUE	AHOLD	DEC/ INC#	BYTE/ WORD#

\* See Section 3.15 for core register addresses.

**BYTE/WORD#** (bit 0). This bit determines whether the FIFO writes words or bytes to the SFB. (See also the BY-32 Mode bit.).

1 = Byte mode. The FIFO writes bytes to the SFB on any byte boundary. The low byte written to *\*outN* is written to the byte address in the SFB. The high byte written to *\*outN* is ignored, and the corresponding byte in the SFB is unchanged.

0 = Word mode. The FIFO writes word-aligned words to the SFB (the address of the low byte must be even).

**DEC/INC#** (Decrement/Increment, bit 1). This bit determines the order in which bytes or words are written to the SFB.

1 = Decrement mode. The FIFO writes from the most significant to the least significant byte or word.

0 = Increment mode. The FIFO writes from the least significant to the most significant byte or word.

**AHOLD** (Address Hold, bit 2). Setting this bit disables automatic incrementing/decrementing of *outN-hi* and *outN-lo*. The FIFO continues to write to a single quad word in the SFB.

**Force LSB Value** (bit 3). The value of this bit is the value of the LSB written to each byte, provided that the FORCE LSB ENABLE bit is set.

**Force LSB Enable**, (bit 4). Setting this bit forces the LSB of each byte written to the SFB to be a '1' or a '0', as specified by the FORCE LSB VALUE bit.

**BY-32 MODE** (bit 5). This bit affects operations only for AHOLD = 0.

1 = The SFB pointer increments or decrements by four bytes.

0 = The SFB pointer increments or decrements by two bytes.

Figure 3-3 shows the data words written to the SFB for the four combinations of the WORD/BYTE and BY-32 MODE bits. These sequences are the same as those for the core input FIFO (see Figure 3-3).

WORD/BYTE = 0 BY-32 MODE = 0	WORD/BYTE = 0 BY-32 MODE = 1
Begin at byte 1.	Begin at byte 1.
"00h"   BYTE 1	"00h"   BYTE 1
"00h"   BYTE 3	"00h"   BYTE 5
"00h"   BYTE 5	"00h"   BYTE 9
WORD/BYTE = 1 BY-32 MODE = 0	WORD/BYTE = 1 BY-32 MODE = 1
Begin at byte 0.	Begin at byte 0.
BYTE 1   BYTE 0	BYTE 1   BYTE 0
BYTE 3   BYTE 2	BYTE 5   BYTE 4
BYTE 5   BYTE 4	BYTE 9   BYTE 8

Figure 3-3. Output FIFO Modes for Writing Data

In BY-32 MODE the pointer increments or decrements by four bytes, regardless of whether the FIFO is in 8-bit pixel mode (WORD/BYTE = 0) or 16-bit pixel mode (WORD/BYTE = 1). BY-32 MODE facilitates writing microcode that operates on one component of an image with 32 bits per pixel. The bytes or words that are skipped are unchanged in the SFB.

The standard sequence for initializing a Core Output FIFO comprises the following steps:

1. Write to the control register (outN-c)



2. Write to the low address register (*outN-lo*)
3. Write to the high address register (*outN-hi*)
4. Write a series of bytes or words to *\*outN*.

There must be one instruction between the write to the Core Output FIFO's low address *outN-lo* and the first write to *\*outN*. Therefore, it is recommended that *outN-lo* be written before *outN-hi* as in the sequence above. Note that this is the reverse of the order for the Input FIFOs.

After writing one or more bytes or words to *\*outN*, and before changing the SFB address, always flush any data that is not yet written to the SFB (i.e., data remaining in *\*outN*). A flush is scheduled by any write to the control register (*outN-c*). To execute the flush, load any valid address into *outN-lo*. The data is lost if you change *outN-hi* or *outN-lo* while data is in the FIFO.

When pointing to a new SFB address that is in the same 64-Kbyte page of the SFB, you need write only the low address to *outN-lo*. However, it is still necessary to have one additional instruction before the first write to *\*outN*.

When writing bytes or words to the SFB through a Core Output FIFO, you can skip a byte or word by writing to *outN++* instead of *\*outN*. When the values are written to the SFB, any byte or word that was skipped retains its original value in the SFB. This can be used when writing a series of pixels, some of which are "transparent," allowing whatever was behind them to show through.

If the microcode routine attempts to write a value to a full Core Output FIFO, the processor is frozen prior to the execution of the write instruction. The processor remains frozen until the FIFO has a chance to write one of the buffered quad words to

the SFB. At that point the processor is released from the frozen state, and the write instruction is executed. When the processor is frozen and waiting for a particular FIFO that is full, that FIFO's SFB access priority is raised above all other FIFOs.

### 3.12 Statistical Decoder

The Statistical Decoder (or Huffman decoder) is a specialized input channel that can read a variable-length bit sequence over the 82750PD internal bus and convert it to a fixed-length bit sequence that is read by the microcode processor. This section discusses the variable-length code, the variable-to-fixed conversion process, and how to control this process with the Statistical Decoder Control Register.

Table 3-10 lists the registers associated with the Statistical Decoder. Further descriptions are given in this section.

Table 3-10. Registers Associated with the Statistical Decoder

Register Mnemonic	Register Name	Description
<i>stat-c</i>	Decoder Control	Specifies the decoding mode, as well as modes for reading and writing the code description table
<i>stat-ram</i>	Decoder Table Write	Writing data to this register stores the data in the code description table
<i>*stat</i>	Decoder Read	Reading this register reads data from the code description table
<i>*stat#</i>	Decoder Hold Read	Reading from this register instead of <i>*stat</i> prevents the decoder from automatically starting to decode the next symbol
<i>stat-lo</i>	Decoder SFB Pointer Low	Least significant byte of the decoder's pointer to the SFB
<i>stat-hi</i>	Decoder SFB Pointer High	Most significant byte of the decoder's pointer to the SFB

### 3.12.1 Statistical Codes

In image compression, as well as in other applications such as text compression, certain data values occur more frequently than others. A means of compressing this data is to use fewer bits to encode more frequently occurring values and more bits to encode less frequently occurring values. This type of encoding results in a variable-length sequence in which the length of a symbol (the group of bits used to encode a single value) can range, for example, from one to sixteen bits.

The statistical code that the decoder can interpret has one of two forms, which are illustrated by the examples in Table 3-11.

Table 3-11. Codes that Can Be Read by the Statistical Decoder

Example of Form A	Example of Form B
0x	1x
10x	01x
110xxx	001xxx
1110xxxx (5 'x's)	0001xxxx (5 'x's)
.	.
.	.
.	.
111111110xxxxx (8 '1's, 6 'x's)	000000001xxxxx (8 '0's, 6 'x's)
1111111110xxxxx (9 '1's, 6 'x's)	0000000001xxxxx (9 '0's, 6 'x's)

Each symbol of a given length (one per line, as in the examples shown in Table 3-11) comprises a run-in sequence followed by some number of x-bits. There are two forms of the run-in sequence. In Form A the run-in sequence consists of zero or more '1's followed by a '0'. In Form B the run-in sequence consists of zero or more '0's followed by a '1'.

The run-in sequence is defined as a series of zero or more '1's followed by a '0' (Form A) or zero or more '0's followed by a '1' (Form B). The remainder of this description uses examples of form A. A bit in the

decoder control register (discussed later) selects the polarity of the run-in sequence bits.

Each x-bit can be '0' or '1'. In this example of Form A there are two symbols of length two: 00 and 01. In general, the number of x-bits on a line in Table 3-11 can be independently set to a value from zero to six. The goal, in general, is to have a few short codes and a larger number of long codes. Thus, codes with fewer run-in bits typically have fewer x's following. However, this is not a hardware constraint.

### Code Description Table

A code of this form is completely described by a code description table. For each length of run-in sequence, this table lists:

- R = the number of '1's in the run-in sequence.
- x = the number of x-bits following the '0'.

The value of R is used as an index to the code description table. However, instead of storing x in the table, we store  $2^x$  because it is easier to implement in logic.

For the example above, the corresponding code description table is shown in Table 3-12.

Table 3-12. Sample Code Description Tables

x	R	$2^x$ (dec.)	$2^x$ (bin.)
1	0	2	000 0010
1	1	2	000 0010
3	2	8	000 1000
5	3	32	010 0000
	...		
6	7	64	100 0000

Note that the highest number of '1's in the run-in is seven. For symbols with more than seven '1's, the value of  $x$  and  $2^x$  for seven '1's is used for all symbols having seven or more '1's in the run-in sequence. For example, in the code above a symbol with eight or more '1's in the run-in sequence has six  $x$ -bits following the '0', which is the same as for symbols having seven '1's.

For each different symbol, including all symbols of the same run-in length with different  $x$ -bit values, the decoder generates a unique fixed-length, 16-bit value. Some of the decoded values for the sample code given above are listed in Table 3-13.

Table 3-13. Decoded Values

Symbol*	Decoded Value
00	0
0 <u>1</u>	1
10 <u>0</u>	2
10 <u>1</u>	3
110 <u>000</u>	4
110 <u>001</u>	5
110 <u>010</u>	6
...	...
110 <u>111</u>	11
1110 <u>00000</u>	12
...	...
1110 <u>11111</u>	43
...	...

\* The  $x$ -bits of the symbol are underlined for clarity.

The following steps comprise an algorithm for generating a decoded value from a symbol:

1. All symbols of a given run-in length are assigned a base value B, which is given by the following formula:

$$B(R) = \text{SUM}[2^{x(r)}] \text{ with } r = 0, 1, 2, \dots, R-1$$

where  $x(r)$  corresponds to the x-value in the table entry corresponding to  $R = r$  (see Table 3-12).

2. The value corresponding to a particular symbol is equal to B plus the binary value of the x-bits in the symbol.

For example, in the above code:

$B(0) = 0$ . ( $B(0)$  is always '0')

$B(1) = 0 + 2 = 2$

$B(2) = 0 + 2 + 2 = 4$

$B(3) = 0 + 2 + 2 + 8 = 12$

$B(4) = 0 + 2 + 2 + 8 + 32 = 44$

Note that a logical implementation of  $B(R)$  is easier as a summation of  $2^{x(r)}$  than it would be for a summation of  $x(r)$ . This is a reason for using  $2^x$  in the code description table (Table 3-12).

#### Short Mode and End Mode

The implementation of this coding scheme in the 82750PD core can be enhanced by using two modifications, SHORT mode and END mode.

SHORT mode allows the decoder to be switched easily to a simpler code format without having to reload the code description table. In SHORT form all symbols have the same number of x-bits, as though all entries in the table had been filled with the same

value of  $2^X$ . SHORT mode is invoked by setting the SHORT bit in the Statistical Decode Control Register (discussed in the next section), and the value of  $2^X$  is placed in the SVAL field of this register (see Table 3-17 below).

END mode consists of dropping the '0' at the end of the longest run-in sequence. For example, consider the code:

0  
10x  
110x

END mode shortens the last symbol to 11x instead of 110x. The trailing '0' is not required because the decoder is told that the maximum length of a run-in is two '1's. The resulting symbol set and corresponding decoded values are given in Table 3-14.

Table 3-14. Decoded Values Using END Mode

Symbol	Decoded Value
0	0
100	1
101	2
110	3
111	4

In END Mode the number of x-bits must be constant for all symbols of the same run-in length. Therefore a code such as the following is **incorrect**:

0  
10xx  
11xxx

The last symbol (11xxx in this case) uses the same table entry for  $2^X$  as the next-to-last symbol (10xx) and, therefore, the last symbol should be 11xx.

The maximum length of the run-in sequence in END mode is specified by placing an END flag in the code description table. For example, a code and the corresponding table are shown in Table 3-15.

Table 3-15. END Flag Decoded Values

Code	Table Entries		
	Index	END bit	$2^X$
0	0	0	0
10xx	1	0	4
110xxx	2	1	8
111xxx	3	—	—
	4	—	—
	5	—	—
	6	—	—
	7	—	—

The hyphens indicate that those table entries are not used to decode this code. Note that the symbol 111xxx has three x-bits because of the value of  $2^X$  in Index 2: it is not based on the  $2^X$  value in Index 3.

SHORT mode and END mode can be invoked simultaneously, resulting in a code such as:

0x  
10x  
110x  
111x



SHORT mode provides a value of  $2^x$  equal to 2 (for one x-bit in each symbol). The END bit is set for Index = 2.

The Statistical Decoder can read packed binary fields, with one to seven bits per field, by:

- setting the END bit in Index 0, and
- programming the x value to be  $N-1$ , where  $N$  is the number of bits per field.

For example, packed three-bit fields could be decoded as shown in Table 3-16.

Table 3-16. Packed 3-bit Field Decoded Values

Code	Table Entries		
	Index	END bit	$2^x$
0xx	0	1	4 ( $N = 3$ so $x = 2$ )
1xx	1	—	—
	2	—	—
	3	—	—
	4	—	—
	5	—	—
	6	—	—
	7	—	—

The order of the unpacked bits is the reverse of the order in the SFB. For example, if three-bit values are packed in the SFB, the pattern 110 in the SFB is read from right to left and gives an unpacked or decoded value of 3.

### 3.12.2 Statistical Decoder Control Register

Table 3-17 describes the Statistical Decoder Control Register (stat-c), which specifies the decoding mode

as well as modes for reading and writing the code description table.

Table 3-17. Statistical Decoder Control Register

Mnemonic: <i>stat-c</i>				Address: Core Register*
Access: R/W				Reset State: Not available.
Bit No.	15	14	13	12-8
Name	POL	RSVD	CB	SVAL

\* See Section 3.15 for core register addresses.

Bit No.	7	6	5	4	3	2-0
Name	SHORT	END	TEST	WRITE	RSVD*	STNDX

**STNDX** (Starting Index, bits 2-0). These bits specify the starting index value of the code description table. The table access begins at this value.

**WRITE** (bit 4) and **TEST** (bit 5). These bits control reads and writes to the code description table, as shown in Table 3-18.

Table 3-18. Control Bits for the Code Description Table.

TEST	WRITE	Function
0	0	After writing new values to the code description table, clear <b>WRITE</b> and <b>TEST</b> to activate the new values.
0	1	Enables a write to the code description table.
1	0	Enables a read of the code description table.
1	1	Reserved

**END** (bit 6). Setting this bit invokes **END** mode. The '1'/'0' trailing the run-in sequence is dropped from the code.

**SHORT** (bit 7). Setting this bit invokes SHORT mode. The values of  $2^X$  are the same for every entry in the code description table.

**SVAL** (Short Value, bits 12-8). If the SHORT bit is set, these bits specify the value of  $2^X$  that is used in the code description table in SHORT mode.

**CB** (Circular Buffer, bit 13). Setting this bit enables the creation of circular buffers of size 64 Kbytes, 128 Kbytes, or 256 Kbytes. Table 3-7 shows the sizes specified by the *circbuf* register.

**POL** (Polarity, bit 15). This bit specifies the polarity of the bits in the run-in sequence.

### 3.12.3 Writing and Reading the Code Description Table

To write to the code description table, execute the following sequence:

1. Set the WRITE bit (bit 4) to enable writes to the table.
2. Set up a pointer to point to the table index where you want to write.
3. Write the table entry to the *star-ram* register. Each 8-bit entry consists of the END bit in bit 7 and the value of  $2^X$  in bits 6-0. Each write to *star-ram* increments the index by one. The index wraps around from seven to zero.

To read the code description table, execute the following sequence.

1. Set the TEST bit (bit 4) of the *c-star* register.
2. Read the table entries from the decoder's Data Register (*\*stat*). Reads and writes always start at table entry zero.

**NOTE:**

*When reading the code description table, it is necessary to wait one instruction time between the write to stat-c and the first read from the \*stat register.*

The example below illustrates the insertion of one instruction time in the code for reading the table entries in the first eight locations of data RAM.

```

dram3 = 0 stat-c = 0x20
cnt = 8
LOOP:
    *dram3++ = *stat cnt--
jcp LOOP

```

Setting the END bit (bit 6) in stat-c enables END mode. Setting the SHORT bit enables SHORT mode. When the decoder is in SHORT mode, the five SVAL bits (bits 12-8) in the CONTROL register are the value of  $2^X$  to be used in all table entries.

The POL bit (bit 15) determines the polarity of the bits in the run-in sequence. Setting POL selects a run-in sequence of '1's ending in '0' (e.g., 1110xxx). Clearing POL selects a run-in sequence of '0's ending in '1' (e.g., 0001xxx).

To set up a circular buffer in the SFB, set the CB bit (bit 13). Then, bits 2-0 of the circular buffer register determine the buffer size (as shown in Table 3-7) and the external address pin that is forced to '0'. (Register pointer unchanged.).

The decoding parameters can be changed between symbols by writing to the stat-c register and, if necessary, writing new values into the code description table. The procedure for changing the code type or decode mode is to read the last value from the decoder prior to the change using \*stat#

instead of *\*stat*. This keeps the decoder from automatically starting to decode the next symbol. At this point, the code description table and the SHORT and END mode bits can be changed as desired. The next time the *stat-c* register is written with both TEST = 0 and WRITE = 0, the decoder begins to decode the next symbol using the new parameters.

The Statistical Decoder buffers one quad word that is read over the 82750PD internal bus. As a result, the decoding of bits in one 32-bit word and the fetch of the next 32-bit word may overlap. As with the core FIFOs, the decoder has an associated SFB pointer that points to the location in the SFB from which it is reading data. This pointer increments twice each time a new quad word is read; there is no decrement mode. When the least significant word of the decoder's pointer (*stat-lo*) is written, any data that had previously been prefetched over the 82750PD internal bus is ignored, and the decoder fetches one quad word starting from this new location.

The 82750PD core assumes that the statistically encoded bit-stream in the SFB starts with the least significant bit of a *double* word. That is, the two LSBs of the address written to *stat-lo* are ignored.

The Statistical Decoder decodes data at a rate of one bit per clock cycle. To a first approximation, the decode time for an N-bit symbol is:

$$\text{decode time} = N + 1 \quad (\text{clock cycles})$$

Since decoding data from one quad word takes at least 64 clock cycles, which is the time required for eight quad word reads over the 82750PD internal bus, the decoder should rarely run out of data. Therefore, the above estimate of the decoding rate should be accurate.

The Statistical Decoder always begins reading the bit-stream from the least significant bit of the dword at the starting location in the SFB. That is, the decoder, unlike the core FIFOs, can start only on a dword boundary. The bit stream moves from the least significant bit to the most significant bit of a dword and then to the least significant bit of the next dword (at the next higher address location). For the x-bits, the first x-bit read from the bit-stream becomes the most significant bit of the x-bit field when it is interpreted as a binary number.

The example below shows a code definition, a bit-stream stored in the SFB, and the resulting decoded values. The code definition and the range of values for each symbol are indicated in Table 3-19.

**Table 3-19. SFB Bit-Stream Decode Values**

Symbol	Values	Comments
0	-	
10x	1, 2	100 = 1, 101 = 2
110xx	3-6	11000 = 3 .... 11011 = 6
1110xxx	7-14	1110000 = 7    1110111 = 14

Decoding starts at address 0 in this example. The double words at addresses 0 and 1 are:

0: 0AC98E14Dh

1: 0372E74CBh

The bit-stream in the SFB, with colons dividing the symbols (read from right to left starting at LSB of address 0), is shown in Figure 3-4. Table 3-20 lists the symbols in the order they are encountered in the bit stream, and the corresponding decoded values.

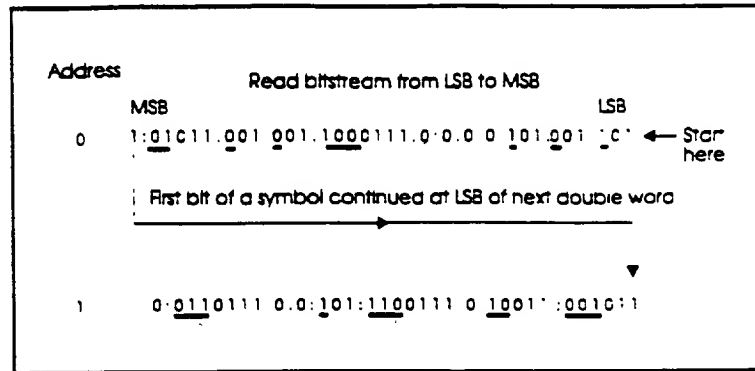


Figure 3-4. SFB Bit-Stream Decoding Addresses

Table 3-20. Decoding Symbols

Symbol	Value	Comments
101	2	Starts at LSB Address 0 scanning left
100	1	
101	2	
0	0	
0	0	
0	0	
0	0	
1110001	8	
100	1	
100	1	
11010	5	
1110100	11	Spans 1st and 2nd dword
11001	4	
0	0	
1110011	10	
101	2	
0	0	
0	0	
1110110	13	
...	...	

### 3.13 Pixel Interpolator

The Pixel Interpolator performs a bilinear interpolation on four 8-bit pixels to generate, in effect, a pixel shifted by a fraction of a pixel position. Figure 3-5 shows four pixels with values A, B, C, and D and the interpolated pixel with value W. The horizontal and vertical weightings are h and v, respectively. The interpolated value, ignoring any quantization effects, is then given by:

$$W = A(1-h)(1-v) + B h(1-v) + C(1-h)v + D h v$$

The values of h and v are even multiples of 1/16. In Figure 3-5 the horizontal weighting is 6/16 and the vertical weighting is 10/16.

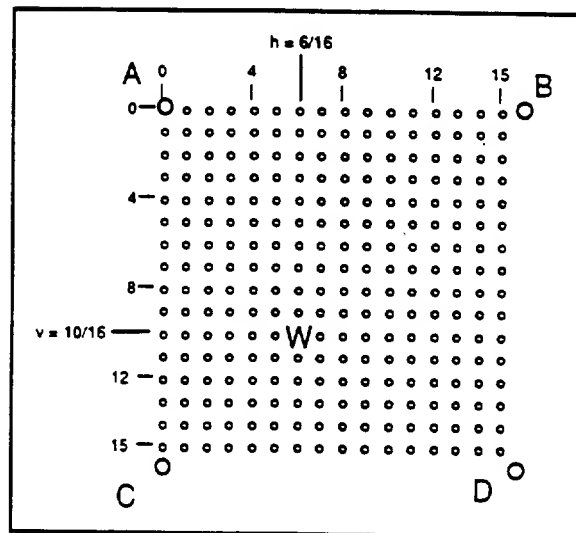


Figure 3-5. Pixel Interpolation



The Pixel Interpolator can operate in two modes: Sequential-2D and Random-2D. Sequential-2D mode is used for motion video decoding and when an array of pixels is interpolated with a common weighting. Random-2D mode is used either when the pixel arrays to be interpolated are not adjacent pixels in two rows or when the weight is changed for each interpolation. ("Random" is used here to mean non-sequential.)

### 3.13.1 Sequential Interpolation

Figure 3-6 illustrates Sequential-2D interpolation. A single row of pixels (W, X, Y, Z, ...) is interpolated using two rows from the original (source) bitmap. The h and v weightings are constant for all of the interpolated pixels.

A	B	E	F	I	...	First input row
	W	X	Y	Z		... Interpolated row
C	D	G	H	K	...	Second Input row

Figure 3-6. Sequential-2D Pixel Interpolation

Source pixels are written to the interpolator as pixel pairs. In Figure 3-6, the pixel pair BA would be written first, followed by the pixel pair DC. (It may seem more natural to refer to the pixel pair as AB, but because of the way 8-bit pixels are arranged in 16-bit words in the SFB, the left-most pixel on the screen is in the least significant byte position. For example, if pixel A had a hex value of 0AAh, and B had a value of 0BBh, the value of the 16-bit word containing pixels A and B would be 0BBAAh.)

The Pixel Interpolator is pipelined and requires a startup sequence to fill the pipeline. Once filled, the interpolator generates a new interpolated pixel every two clock cycles in Sequential-2D mode.

After the pairs BA and DC have been written, two pixels are read from the interpolator. Because the pipeline is not yet full, these pixels are read and discarded. This loop of writing two pixel pairs and reading two output pixels continues four times. The two pixels that are read the fourth time are the first two valid output pixels: W and X. The interpolator may also collect output (interpolated) pixels into a 16-bit pixel pair XW. There are two possible phase relationships between the input pixel pairs and output pixels pairs (either X and W paired or Y and X paired). Either phase can be specified for the interpolator.

### 3.13.2 Random Interpolation

Random-2D interpolation is used either when the pixels to be interpolated are not in horizontal rows or when the weight is changed for each interpolated pixel. In Random-2D mode, the processing for successive interpolated pixels cannot take advantage of pipelining: each pixel is considered to be the first pixel in a Sequential-2D interpolation. The weight and the two input pixel pairs are written into the interpolator. After waiting at least ten clock cycles (in standard mode) or six clock cycles (in fast mode), the single interpolated pixel can be read. Then, the next two input pixel pairs and, if necessary, the new weight value, are written. Ten clock cycles later the next interpolated pixel can be read.

### 3.13.3 Pixel Interpolator Control

Table 3-21 shows the Pixel Interpolator Control Register (*pixint-c*). Following paragraphs discuss the bit combinations for different modes of interpolation.

Table 3-21. Pixel Interpolator Control Register

Mnemonic: *pixint-c*

Address: Core Register\*

Access: R/W

Reset State: Not available.

Bit No.	15	14	13	12	11	10	9-8
Name	RSVD	PIPELINING	PHASE	RSVD	PAIRING	RESET	MODE SELECT

\* See Section 3.15 for core register addresses

Bit No.	7-4	3-0
Name	VERTICAL WEIGHT	HORIZ WEIGHT

**HORIZ WEIGHT** (Horizontal Weighting, h, bits 3-0).  
**VERTICAL WEIGHT** (Vertical Weighting, v, bits 7-4).  
 These bits contain the horizontal and vertical weightings, expressed as the numerator of a fraction that is an even multiple of 1/16, i.e., h and v must be chosen from the following values: 0, 2, 4, 6, ..., 14.

**MODE SELECT** (Mode Selection, bits 9-8). These bits select the operating mode, as shown in Table 3-22. Use the Random-2D mode if the pixels are not in horizontal rows or if the h and v weightings are not constant over all pixels. Use the Sequential-2D mode if the h and v weightings are constant for all interpolations.

Table 3-22. Interpolator Mode Selection

Bit 9	Bit 8	Operating Mode
0	0	Random-2D
0	1	Sequential-2D
1	0	Reserved
1	1	Reserved

Note: The interpolator must be reset (set the RESET bit) when switching modes.

**RESET** (Reset, bit 10). Setting this bit resets the Pixel Interpolator by flushing the source pixels stored in the interpolator. Set this bit when making a mode change (writing to the MODE SELECT bits.)

**PAIRING** (Pixel Pairing, bit 11). This bit determines whether the interpolator outputs a single pixel or a pair of pixels.

1 = The interpolator outputs 16-bit pixel pairs comprising adjacent pixels, provided that the MODE SELECT bits select Sequential-2D mode. When combined with the ALU's dual-add-with-saturate operation, this feature aids in motion video decoding by allowing two pixels to be processed during each cycle.

0 = The Pixel Interpolator outputs individual 8-bit pixels.

**PHASE** (Bitmap Phase, bit 13). This bit selects the phase (alignment) of the output pixel pairs relative to the input pixel pairs, as shown in Figure 3-7. (Recall that the pixel on the left is the least significant pixel of a pair. For example, A is the least significant pixel of the pair BA.

1 = Out of phase. The first pixel (W) is placed in the most significant byte of the first pixel pair; the least significant byte of the first pixel pair contains invalid data.

0 = In phase. The first two output pixels (W and X) are grouped as one 16-bit pair with the first pixel (W) in the least significant byte.

This bit also affects the pipeline delay (see the PIPELINING bit).

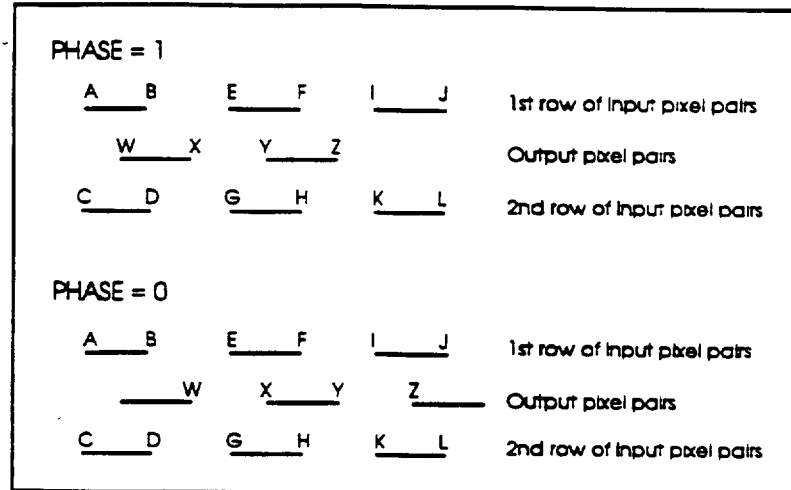


Figure 3-7. Pixel Pair Phases

**PIPELINING.** (Pipelining, bit 14). Setting this bit reduces the pipeline delay, which is determined by this bit and the PHASE bit, as shown in Table 3-23.

Setting the PIPELINING bit selects a faster mode, which has reduced pipeline delay compared to the standard pipelining mode (PIPELINING = 0).

Table 3-23. Pipeline Delay

PIPELINING (bit 14)	PHASE (bit 13)	Pipeline delay (in output pixels)
0	0	6
0	1	-
1	0	2
1	1	3

Changing the PAIRING bit (bit 11) from '0' (single pixels) to '1' (pixel pairs) does not change the amount of pixel delay, but half as many reads and writes are required to fill the pipeline because each read or write transfers two pixels. For example, in standard mode (PIPELINING = 0), with zero phase (PHASE = 0) and pair mode (PAIRING = 1), three indeterminate pixel pairs must be read before the first valid pixel pair is read. In the same case, but with the Phase bit = 1, the fourth pixel pair read contains one valid pixel and one indeterminate pixel, and the fifth pixel pair read contains two valid pixels.

### 3.14 Control, Status, and Interrupt Flag Registers

Three 82750PD core registers control and monitor the 82750PD interrupt and other core operations.

The 82750PD has four interrupt sources, which share a single interrupt line to the host CPU. Any one of the four sources (if enabled) can assert the single 82750PD interrupt. For convenience, we refer to the four interrupt sources as the four "82750PD interrupts," with the understanding that these are combined to form the single interrupt to the host.

In addition to the four 82750PD interrupts described in this section, there are *meta-interrupts* associated

with the SynchroLink. Meta-interrupts are described in Section 6.5. Section 5.7 describes the combined interrupt and meta-interrupt system and gives a procedure for setting up the 82750PD interrupt to the host.

This section describes the following three core registers:

- The Core Control Register (*ccontrol*, W/O) has bits for enabling the four 82750PD interrupts and bits to control other aspects of core operation.
- The Core Interrupt Flag Register (*cingflag*, R/O) has bits that indicate which interrupts are asserted.
- The Core Status Register (*cstatus*, R/O) has bits that reflect the states of the interrupt enabling bits in the *ccontrol* register and other status bits.

The next three subsections cover these registers individually. Section 3.14.4 discusses the four 82750PD interrupts and the functions of the registers in the interrupt system.

### 3.14.1 Core Control Register

---

Table 3-24 shows the Core Control Register (*ccontrol*). This write-only register has bits that control the operation of the core and bits that enable the 82750PD interrupt sources. Setting an enable bit (*x\_E*) enables the corresponding interrupt condition to assert the 82750PD interrupt. Section 3.14.4 describes the individual interrupt sources.

Table 3-24. Core Control Register

Mnemonic: *ccontrol*

Byte Offset: 100h

Access: W/O

Reset State: 01h

Bit No.	15	14	13	12	11-8
Name	CORE_EN	RSVD	PMON/FRZ	1*	RSVD

Bit No.	7	6	5-4	3	2	1	0
Name	EFI_E	OFI_E	RSVD	VBI_E	MCINT_E	STEP	HALT

\* Bit 12 must be set when writing to this register

**HALT** (Halt, bit 0).**STEP** (Step, bit 1).

The combined values of the HALT and STEP bits command the core to run normally, to execute a single instruction, or to halt, as shown in Table 3-25. (Note that the CORE\_EN bit must be set for the core to function in any of these modes.)

Table 3-25. Run, Step, and Halt Modes.

HALT	STEP	Mode
0	0	Core runs normally
0	1	Core runs normally
1	0	Core halts
1	1	Core executes one instruction and then halts

**MCINT\_E** (Microcode Interrupt Enable, bit 2).**VBI\_E** (Vertical Blanking Interrupt Enable, bit 3).**OFI\_E** (Odd Field Interrupt Enable, bit 6).**EFI\_E** (Even Field Interrupt Enable, bit 7).

Setting one of these bits enables the corresponding interrupt condition to assert the 82750PD interrupt. Only one enable bit of VBI\_E, OFI\_E, and EFI\_E should be set at a time. Section 3.14.4 has a further discussion of these bits.



**PMON/FRZ** (Performance Monitor/Freeze, bit 13). This bit determines which output signal is on the PMFRZ# pin. This bit is functional only for ISA and PCI buses.

1 = Output signal FRZ# is on PMFRZ# pin.

0 = Output signal PMON# is on PMFRZ# pin.

**CORE\_EN** (Core Enable, bit 15). This bit must be set to enable the core to run in any of the modes as determined by the HALT and STEP bits.

### 3.14.2 Core Interrupt Flag Register

The Core Interrupt Flag Register (*cinflag*) has a flag bit for each of the four interrupt sources in the 82750PD core (see Table 3-26). A flag bit is set when the interrupt condition is detected (regardless of the state of the corresponding interrupt Enable bit in the *ccontrol* register). Reading the *cinflag* register clears all of the flags. If the *cinflag* register is read and an interrupt condition is detected during the same cycle, the flag bit corresponding to that interrupt condition is set. This new interrupt condition is then seen by the host processor when it next reads the *cinflag* register. The flag ensures that an interrupt condition is not lost if it occurs at the same cycle that the *cinflag* register is read (and reset). In addition, the microcode interrupt source has an overflow flag (MCINTO\_F) that is set if more than one microcode interrupt condition has occurred since the *cinflag* register was last read. Further descriptions of these bits are in Section 3.14.4.

Table 3-26. Core Interrupt Flag Register

Mnemonic: *cntflag*

Byte Offset: 100h

Access: R/O

Reset State: 1FFh

Bit No.	15	14	13	12	11	10	9	8
Name	RSVD	VBI_F	MCINT_F	RSVD	MCINTO_F	OFI_F	EFI_F	RSVD

Bit No.	7—0
Name	RSVD

**MCINT\_F** (Microcode Interrupt Flag, bit 13).**VBI\_F** (Vertical Blanking Interrupt Flag, bit 14).**OFI\_F** (Odd Field Interrupt Flag, bit 10).**EFI\_F** (Even Field Interrupt Flag, bit 9).

Any bit in this group is set when its corresponding interrupt condition is detected, regardless of the state of the enable bit in the *ccontrol* register.

### 3.14.3 Core Status Register

The 82750PD Core Status Register (*cstatus*) provides status information on the 82750PD core (see Table 3-27). Four of the bits (MCINT\_S, VBI\_S, OF\_S, and EF\_S) reflect the values of the corresponding *ccontrol* register bits that enable the interrupt sources. Note, however, that the bit positions and bit ordering are not the same as in the *ccontrol* register.

Table 3-27. Core Status Register

Mnemonic: *cstatus*

Byte Offset: 102

Access: R/O

Reset State: 1FFh

Bit No.	15	14	13-12	11	10	9-8
Name	EFL_S	OFL_S	RSVD	VBI_S	MCINT_S	RSVD

Bit No.	7-3	2	1	0
Name	RSVD	PMON	FREEZE	HALT_S

**HALT\_S** (Halt Status, bit 0). This bit is set when the processor is halted due to at least one of the following conditions:

- The HALT bit in the *ccontrol* register is set.
- The HALT bit in the INT CFG register is set (see Section 5.8).
- The HALT# pin is asserted.

**FREEZE** (Freeze, bit 1). This bit is set when the processor is waiting for at least one of the following:

- The Statistical Decoder.
- Data to be fetched from the SFB for an input FIFO.
- Data from an output FIFO to be written to the SFB.

**PMON** (Performance Monitor, bit 2). This bit, which can be toggled by the ALU "performance monitor" operation, can be used to monitor microcode performance.

**Bits 3-9.** Reserved.

**MCINT\_S** (Microcode Interrupt Status, bit 10).  
**VBI\_S** (Vertical Blanking Interrupt Status, bit 11).  
**OFI\_S** (Odd Field Interrupt Status, bit 14).  
**EFI\_S** (Even Field Interrupt Status, bit 15).  
Each of these bits reflects the state of its corresponding bit in the *ccontrol* register. Note that the corresponding bits in *ccontrol* have different bit numbers.

#### 3.14.4 Summary of Interrupt Bits

Table 3-28 lists the four 82750PD interrupt conditions, the bit name prefixes for the interrupt conditions, and the bit numbers in the *intflag*, *ccontrol*, and *cstatus* registers.

Hardware sets a flag bit in the *intflag* register upon detection of the corresponding interrupt condition. Software sets an enable bit in the *ccontrol* register to enable the corresponding interrupt condition to actually assert the 82750PD interrupt. Only one of the VBUS code interrupts (VBI, OFI, EFI) should be enabled at a time.

Hardware sets a status bit in the *cstatus* register when the corresponding bit in the *ccontrol* register is set by software. (Note that the control, status, and flag bits for a specific interrupt are in different bit positions in the three registers.)

Table 3-28. Interrupt Bits in the Flag, Control, and Status Registers

Interrupt Condition	Bit Prefix	Flag* Bit No. in <i>cintflag</i>	Enable** Bit No. in <i>ccontrol</i>	Status*** Bit No. in <i>cstatus</i>
Microcode Interrupt: The MCINT interrupt is generated by the "Interrupt Host" ALU operation	MCINT	13	2	10
Vertical Blanking Interrupt: The VBI interrupt is generated upon detection of either a VODD or a VEVEN VBUS code	VBI	14	3	11
Odd Field Interrupt: The OFI interrupt is generated upon detection of a VODD VBUS code	OFI	10	6	14
Even Field Interrupt: The EFI interrupt is generated upon the detection of a VEVEN VBUS code	EFI	9	7	15

\* Hardware sets this bit when an interrupt condition is detected.

\*\* Setting this bit enables an interrupt condition to interrupt the host CPU

\*\*\* This bit reflects the state of the corresponding event in the *ccontrol* register

### 3.15 Host Access to the Core Registers

This section describes how the host computer accesses the core registers. The host accesses the core registers at addresses 00FE0000h—00FEFFFFh in the 82750PD memory address space, as described in Section 2.5. Note that the 82750PD core can access these registers only in microcode. The access information in this section does not apply to the core.

Table 3-29 lists the core register addresses in terms of offsets from the base address for host accesses to the core. The base address is 00FE0000h.

Table 3-29. Host Address Mapping of the Core Registers

Byte Address*	Register Category
(a) 000h—07Eh	A bus source and destination registers
(b) 080h—0FEh	B bus source and destination registers
(c) 100h—17Eh	Microcode processor status and control registers
(d) 180h—1FEh	SFB pointer locations in RAM

\* Offsets from the base address 00FE 0000h

**NOTE:**

*The host may address the core registers using word-aligned word accesses only. Software is responsible for enforcing this restriction.*

When the host CPU accesses areas (a), (b), or (d) in Table 3-29, and the 82750PD is not already in a Halt state, the 82750PD automatically halts for the one clock cycle actually required to complete the data transfer. The 82750PD then restarts. If the 82750PD is in a Halt state when the host access is initiated, it remains in the Halt state following the access. This is transparent to both the host and the microcode processor.

During an access to areas (a) or (b), bits 6–1 of the byte offset should contain the source or destination byte address for the register to be accessed. Bit 0 is always clear. The source/destination addresses are given in Tables 3-30 and 3-31.

Table 3-32 lists the byte addresses for the SFB pointers.

Table 3-30. 82750PD A Bus Source/Destination Address

Address (Hex)	ADST	ASRC	Address (Hex)	ADST	ASRC
000h	nu!		040h	*out1	*in1
002h		hwid	042h	out1—	*in2
004h		cc	044h	shift-r	*stat
006h	maddr		046h	out1-hi	*stat#
008h		alu	048h	*out2	
00Ah	cnt	cnt	04Ah	out2—	
00Ch	cnt2	cnt2	04Ch	shift-r	
00Eh	lcnt	lcnt	04Eh	out2-hi	
010h	r0	r0	050h	out1-c	
012h	r1	r1	052h	in1-c	
014h	r2	r2	054h	shift-1	
016h	r3	r3	056h	in1-hi	
018h	r4	r4	058h	out2-c	
01Ah	r5	r5	05Ah	in2-c	
01Ch	r6	r6	05Ch		
01Eh	r7	r7	05Eh	in2-hi	
020h	mcode3	mcode3	060h	r8	r8
022h	mcode2	mcode2	062h	r9	r9
024h	mcode1	mcode1	064h	r10	r10
026h	pc	pc	066h	r11	r11
028h	pixint-c		068h	r12	r12
02Ah	pixint	pixint	06Ah	r13	r13
02Ch	*dram1	*dram1	06Ch	r14	r14
02Eh	*dram2	*dram2	06Eh	r15	r15
030h	*dram1++	*dram1++	070h	cc	shift
032h	*dram2++	*dram2++	072h	lcnt	lcnt
034h	*dram1--	*dram1--	074h	*dram3	*dram3
036h	*dram2--	*dram2--	076h	*dram4	*dram4
038h	dram1	dram1	078h	*dram3--	*dram3--
03Ah	dram2	dram2	07Ah	*dram4--	*dram4--
03Ch	dram3	dram3	07Ch	*dram3--	*dram3--
03Eh	dram4	dram4	07Eh	*dram4--	*dram4--

Table 3-31. 82750PD B Bus Source/Destination Address

Address (Hex)	BDST	BSRC	Address (Hex)	BDST	BSRC
080h	nut		0C0h	*out1	prof
082h		alu	0C2h	out1---	
084h	*dram3	*dram3	0C4h	out1-lo	out1-lo
086h	*dram4	*dram4	0C6h	out1-hi	out1-hi
088h	*dram3++	*dram3++	0C8h	*out2	stat-lo
08Ah	*dram4--	*dram4--	0CAh	out2---	stat-hi
08Ch	*dram3--	*dram3--	0CCh	out2-lo	out2-lo
08Eh	*dram4--	*dram4--	0Ceh	out2-hi	out2-hi
090h	r0	r0	0D0h	out1-c	out1-c
092h	r1	r1	0D2h	in1-c	in1-c
094h	r2	r2	0D4h	in1-lo	in1-lo
096h	r3	r3	0D6h	in1-hi	in1-hi
098h	r4	r4	0D8h	out2-c	out2-c
09Ah	r5	r5	0DAh	in2-c	in2-c
09Ch	r6	r6	0DCh	in2-lo	in2-lo
09Eh	r7	r7	0DEh	in2-hi	in2-hi
0A0h	r8	*in1	0E0h	stat-ram	r8
0A2h	r9	*in2	0E2h	stat-c	r9
0A4h	r10	*stat	0E4h	stat-lo	r10
0A6h	r11	*stat#	0E6h	stat-hi	r11
0A8h	r12	circbuf	0E8h	yeven-lo	r12
0AAh	r13		0EAh	yeven-hi	r13
0ACh	r14		0ECh	yodd-lo	r14
0AEh	r15		0EEh	yodd-hi	r15
0B0h	circbuf	literal 0	0F0h	ypitch	shift
0B2h		literal 1	0F2h		stat-c
0B4h	*dram1	literal 2	0F4h	vu-lo	*dram1
0B6h	*dram2	literal 3	0F6h	vu-hi	*dram2
0B8h	*dram1--	literal 4	0F8h	vupitch	*dram1--
0BAh	*dram2--	literal 5	0FAh	vpitch	*dram2--
0BCh	*dram1--	literal 6	0FCh	vptr-lo	*dram1--
0BEh	*dram2--	literal 7	0FEh	vptr-hi	*dram2--



Table 3-32. SFB Pointer Addresses

Byte Address	Mnemonic	Description
180 182		Reserved
184h 186h	out1-lo out1-hi	Output FIFO 1 Pointer
188h 18Ah		Reserved
18Ch 18Eh	out2-lo out2-hi	Output FIFO 2 Pointer
190h 192h		Reserved
194h 196h	in1-lo in1-hi	Input FIFO 1 Pointer
198h 19A		Reserved
19Ch 19Eh	in2-lo in2-hi	Input FIFO 2 Pointer
1A0h 1A2h		Reserved
1A4h 1A6h	stat-lo stat-hi	Working Copy of Statistical Decoder Pointer
018Ah— 01BEh		Reserved

Area (c) in Table 3-29 contains three registers: the Core Control Register (*ccontrol*), the Core Interrupt Flag Register (*cnrflag*), and the Core Status Register (*cstatus*), which are described in Section 3.14. Table 3-33 lists the byte offsets for these registers.

Table 3-33. Byte Offsets for the Area-c Registers

Byte Offset for Host Access	Register
100h (WRITE)	Core Control Register ( <i>cccontrol</i> )
100h (READ)	Core Interrupt Flag Register ( <i>cniflag</i> )
102h	Core Status Register ( <i>cstatus</i> )

## Chapter 4

# Microcode Instruction Format

### 4.1 Overview

The 82750PD core executes instructions from microcode RAM (MRAM), which stores 512 48-bit instructions. This chapter describes the execution sequence and the format of the microcode instructions. For a more extensive guide to microcode programming, see *i750 Pixel Processor Microcode Development Tools User's Guide*, Intel Corporation, 1993 (Order Number: 485407-001).

### 4.2 Instruction Sequencing

Figure 4-1 shows the format of a microcode instruction.

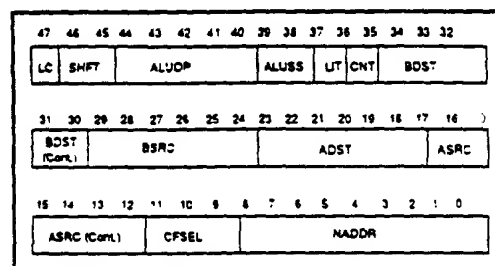


Figure 4-1. Microcode Instruction Format

The bits in the NADDR field determine the address of the next instruction to be executed. The upper eight bits of the field specify a pair of instructions: an odd-address instruction and an even-address

instruction. Each instruction fetch reads both instructions of this pair. One of these is the next instruction to be executed. This selection is described in the next two sections, which describe the NADDR and CFSEL fields.

### 4.3 Instruction Word Field Descriptions

The subsections to follow describe the fields of a microcode instruction.

#### 4.3.1 NADDR -- Next Instruction Address Field

The Next Instruction Address field (NADDR, bits 8-0) holds the address of the next instruction to be executed. A zero-delay, two-way branch can be achieved by taking advantage of the physical organization of the microcode RAM, which is 96 bits wide (the width of two instruction words) by 256 deep. An instruction read cycle fetches a pair of instruction words, one with the even-address and the other with the odd address. The next instruction to be executed (the even-address instruction or the odd-address instruction) is determined by the LSB of the NADDR field of the current instruction and the state of the condition flag. This selection is described in the following section on the CFSEL field.

An instruction that writes to the program counter (pc) is the only type of instruction for which the NADDR field does not determine the next instruction to be executed.

When an instruction (Instruction A) loads the pc, a one-instruction delay occurs before the load takes effect. Therefore, the instruction that follows the write to the pc (Instruction B) is always executed.

However, the processor ignores the NADDR field of Instruction B and jumps to the address in the pc. Table 4-1 shows an example. Note that the instruction following a PC load is **always** executed, even if the processor is in Step mode or if the processor is frozen on that instruction. Another example of writing to the pc register is shown in Appendix C.

Table 4-1. PC Load Example

Address	Instruction	NADDR	Comments
10	pc = 0	55	Load PC with zero
55	r0 = 1	X	This instruction is executed but its next address field is ignored
0	r1 = r0	25	PC load takes effect after a one-instruction delay, the result is that r1 = r0 = 1.

#### 4.3.2 CFSEL -- Condition Flag Select Field

The eight most significant bits of the NADDR field specify an odd-even instruction pair. The LSB of the NADDR field and the Condition Flag Select field (CFSEL, bits 11-9) select the next instruction from this odd-even pair.

The CFSEL field selects the *condition flag*. Table 4-1 lists the condition flag selected for each value in the CFSEL field. These flags are bits in the ALU Condition Code Register (cc, see Section 3.3). A condition flag value is TRUE ('1') or FALSE ('0').

Table 4-2. Condition Flag Select Field Assignments

Value	Flag	Description
000	FALSE	Select this flag for an Unconditional Branch
001	CARRY	Carry Out from ALU Condition Flag Latch
010	ALU OVF	Overflow from ALU Condition Flag Latch
011	ALU SIGN	Sign from ALU Condition Flag Latch
100	ALU ZERO	Zero from ALU Condition Flag Latch
101	LCNTZ	TRUE if Selected Loop Counter = 0
110	LSB	LSB of Data Register <i>r0</i>
111	MSB	MSB of Data Register <i>r0</i>

**NOTE**

*The ALU condition flags (CARRY, OVF, SIGN, and ZERO) are latched in the ALU Condition Flag register. This register is updated for most (but not all) ALU operations. The remaining flags (LCNTZ, LSB, and MSB) are updated and latched each cycle.*

CFSEL = '000' selects the unconditional branch flag. The name of the flag is FALSE because its state is always '0'. The states of the other flags can be '0' (FALSE) or '1' (TRUE).

Table 4-3 shows how the next instruction is chosen from the condition flag state and the LSB of the NADDR field in the current instruction.

Table 4-3. Microcode Next Instruction Selection

LSB of Address	Condition Flag State	Next Instruction
0	0 (FALSE)	EVEN
0	1 (TRUE)	EVEN
1	0 (FALSE)	ODD
1	1 (TRUE)	EVEN

For an unconditional branch (CFSEL = 000), the FALSE condition flag (which is always zero) is selected; this leaves the LSB of the address to determine the next instruction: for LSB = 0 the EVEN instruction is selected, and for LSB = 1 the ODD instruction is selected. This allows unconditional branching to any of the 512 instructions in the microcode RAM.

For a conditional branch (CFSEL  $\neq$  000) the LSB of the address in NADDR is set. In this case (the last two entries in Table 4-3) the state of the condition flag selects the next instruction: FALSE selects the ODD instruction; TRUE selects the EVEN instruction. Therefore, a conditional branch jumps to either the odd or even instruction of an odd/even pair, depending on the state of the condition flag.

#### **4.3.3 ASRC – A Bus Source Select Field**

---

The A Bus Source Select field (ASRC, bits 18-12) selects the unit within the 82750PD core that drives its data onto the A bus during the execution of this instruction. The mapping for this and the following three fields is provided in Table 4-4.

Table 4-4. 82750PD Core Source/Destination Coding

Address (Hex)	BDST	BSRC	ADST	ASRC
00h	nul		nul	
01h		alu		
02h	*dram3	*dram3		cc
03h	*dram4	*dram4	maddr	
04h	*dram3--	*dram3--		alu
05h	*dram4--	*dram4--	cn1	cn1
06h	*dram3--	*dram3--	cn2	cn2
07h	*dram4--	*dram4--	lc1	lc1
08h	r0	r0	r0	r0
09h	r1	r1	r1	r1
0Ah	r2	r2	r2	r2
0Bh	r3	r3	r3	r3
0Ch	r4	r4	r4	r4
0Dh	r5	r5	r5	r5
0Eh	r6	r6	r6	r6
0Fh	r7	r7	r7	r7
010h	r8	*in1	mcode3	mcode3
011h	r9	*in2	mcode2	mcode2
012h	r10	*stat	mcode1	mcode1
013h	r11	*stat#	pc	pc
014h	r12	crcbuf	pixint-c	
015h	r13		pixint	pixint
016h	r14		*dram1	*dram1
017h	r15		*dram2	*dram2
018h	crcbuf	literal 0	*dram1--	*dram1--
019h		literal 1	*dram2--	*dram2--
01Ah	*dram1	literal 2	*dram1--	*dram1--
01Bh	*dram2	literal 3	*dram2--	*dram2--
01Ch	*dram1--	literal 4	dram1	dram1
01Dh	*dram2--	literal 5	dram2	dram2
01Eh	*dram1--	literal 6	dram3	dram3
01Fh	*dram2--	literal 7	dram4	dram4
020h	*out1	prof	*out1	*in1



Table 4-5. 82750PD Source/Destination Coding (Con't.).

Address (Hex)	BDST	BSRC	ADST	ASRC
021	out1---		out1---	*in2
022	out1-lo	out1-lo	shift-ri	*stat
023	out1-hi	out1-hi	out1-hi	*stat#
024	*out2	stat-lo	*out2	
025	out2---	stat-hi	out2---	
0x26	out2-lo	out2-lo	shift-r	
0x27	out2-hi	out2-hi	out2-hi	
0x28	out1-c	out1-c	out1-c	
029	in1-c	in1-c	in1-c	
02A	in1-lo	in1-lo	shift-l	
02B	in1-hi	in1-hi	in1-hi	
02C	out2-c	out2-c	out2-c	
02D	in2-c	in2-c	in2-c	
02E	in2-lo	in2-lo		
02F	in2-hi	in2-hi	in2-hi	
030	stat-ram	r8	r8	r8
031	stat-c	r9	r9	r9
032	stat-lo	r10	r10	r10
033	stat-hi	r11	r11	r11
034	yeven-lo	r12	r12	r12
035	yeven-hi	r13	r13	r13
036	yodd-lo	r14	r14	r14
037	yodd-hi	r15	r15	r15
038	ypitch	shift	cc	shift
039		stat-c	fcnt	fcnt
03A	vu-lo	*dram1	*dram3	*dram3
03B	vu-hi	*dram2	*dram4	*dram4
03C	vupitch	*dram1↔	*dram3↔	*dram3↔
03D	vpitch	*dram2↔	*dram4↔	*dram4↔
03E	vptr-lo	*dram1--	*dram3--	*dram3--
03F	vptr-hi	*dram2--	*dram4--	*dram4--

#### 4.3.4 ADST -- A Bus Destination Select Field

---

The A Bus Destination Select field (ADST, bits 23-18) selects the element that latches data from the A bus during the execution of this instruction. See ASRC above and Table 4-4.

#### 4.3.5 BSRC -- B Bus Source Select Field

---

The B Bus Source Select field (BSRC, bits 29-24) selects the unit within the 82750PD core that drives its data onto the B bus during the execution of this instruction (see Table 4-4).

#### 4.3.6 BDST -- B Bus Destination

---

The B Bus Destination field (BDST, bits 35-30) selects the element that latches data from the B bus during the execution of this instruction. See BSRC above and Table 4-4.

#### 4.3.7 CNT -- Decrement Loop Counter Bit

---

Bit 36 of the instruction word is the Decrement Loop Counter bit (CNT). If the CNT bit is set, the selected Loop Counter (*cnt* or *cnt2*, as selected by the loop counter select bit, LC) is decremented. The new value of the loop counter and the updated LCNTZ condition flag are not ready until the next instruction cycle. Therefore, in a loop where the loop counter is decremented and tested for zero in the same instruction (typically in a one instruction loop), the start value for the loop counter should be one less than the number of times the loop is to be executed.

### 4.3.8 LIT -- Literal Select Bit

Bit 37 is the Literal Select bit (LIT). When the LIT bit is set, the ASRC and CFSEL fields (bits 17-9) become a single 9-bit literal field. The value in this field is driven onto the least significant 9 bits of the A bus, and zeros are driven onto the upper 7 bits of the A bus. Figure 4-2 shows how the bits from ASRC and CFSEL produce the literal value on the A bus.

A Bus Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bits Forced to '0'	0	0	0	0	0	0	0									
ASRC Bits								A	A	A	A	A				
CFSEL Bits													C	C	C	C

Figure 4-2. Literal Field Mapping onto a Bus

#### NOTE

*A conditional branch and a literal on the A bus are not allowed in the same instruction. A 3-bit literal can be placed on the B bus in any instruction.*

### 4.3.9 SHFT -- Shift Control Field

The Shift Control field (SHFT, bits 46-45) controls the bit shifting and byte swapping logic associated with the r0 register. The encoding of this field is given in Table 4-6.

Table 4-6. SHIFT Control Field Coding

SHIFT	Operation
0 0	No shift or swap operation
0 1	Shift <i>r0</i> right 1 bit position, sign extend
1 0	Shift <i>r0</i> left 1 bit position, zero fill
1 1	Byte swap the value being loaded into <i>r0</i> *

\* Byte swapping only works when *r0* is the destination on the A bus or the B bus. The bytes of the data being loaded into *r0* are swapped, not the bytes of the data already in *r0*. In order to byte swap data already in register *r0*, *r0* must be both a source and destination for either the A or B bus.

#### 4.3.10 ALUSS -- ALU Source Select Bits

The bits in the ALU Source Select field (ALUSS, 39-38) are enable bits for the two ALU input latches connected to the A and B buses. Bit 39 enables the A bus latch; bit 38 enables the B-bus latch. A '1' in either bit position causes the corresponding input latch to latch the 16-bit value on the bus to which it is connected (the A or B bus). A '0' in either bit causes the corresponding latch to hold its current content. This allows the ALU operands either to come from "eavesdropping" on the A or B bus transfers occurring in the current instruction cycle or to be held for multiple instruction cycles in either the A or B input latch.

#### 4.3.11 ALUOP -- ALU Operation Code Field

The ALU Operation Code field (ALUOP, bits 44-40) specifies the ALU instruction to be performed during the current instruction cycle. The encoding of this field is given in Table 4-7, and Section 3.3 discusses some of the opcodes. Normally, at the end of the instruction execution, the result of the ALU operation is latched in the ALU output latch (the *alu*

register). This latch can be a source for either the A bus or the B bus. However, if "no operation" is selected for the ALU operation, the ALU output latch is not latched into the *alu* register; the data is held from the previous instruction. Two additional ALU opcodes, "microcode interrupt" and "performance monitor," do not actually perform ALU operations and, therefore, do not latch the ALU results.

Table 4-7. ALU Operations

Bit Coding	Operation	Bit Coding	Operation
00h	No Operation	010h	$a + b$
01h	Zero	011h	$a - b$
02h	Pass a	012h	$-a + b$
03h	Pass b	013h	2's Complement of a
04h	1's Complement of a	014h	2's Complement of b
05h	1's Complement of b	015h	Increment a
06h	$a \text{ AND } b$	016h	Increment b
07h	$(\text{NOT } a) \text{ AND } b$	017h	Decrement a
08h	$a \text{ AND } (\text{NOT } b)$	018h	Decrement b
09h	$a + b + 1$	019h	Interrupt Host
0Ah	$a \text{ OR } b$	01Ah	Performance Monitor
0Bh	$(\text{NOT } a) \text{ OR } b$	01Bh	Pass a, Don't Latch Flags
0Ch	$a \text{ OR } (\text{NOT } b)$	01Ch	Pass b, Don't Latch Flags
0Dh	$a - b - (\text{Previous Borrow})$	01Dh	$a + b - (\text{Previous Carry})$
0Eh	$a - b$	01Eh	Dual Add with Saturate*
0Fh	$-a + b - (\text{Previous Borrow})$	01Fh	Dual Sub with Saturate*

#### 4.3.12 LC -- LOOP Counter Select Bit

---

The Loop Counter Select bit (LC) selects which of the two loop counters is used for decrementing or Loop-Counter-Zero conditional branching in the current instruction (see Section 3.6).

0 = Select *cnt1*

1 = Select *cnt2*

## Chapter 5

# UNIVERSAL HOST BUS INTERFACE

---

The 82750PD Universal Host Bus Interface (UHBI) is the link between the host CPU and the other components of the multimedia system. The UHBI, which can accommodate different bus types provides the host with four access modes to the other system components: direct I/O, indirect I/O, an EMS-like memory access mode, and a pair of host-SFB FIFOs. In addition, the UHBI supports a single 82750PD interrupt.

### 5.1 Introduction to the Universal Host Bus Interface

---

The Universal Host Bus Interface supports five bus types:

- ISA
- EISA
- Micro Channel
- PCI
- VL-Bus

The host can access the multimedia system in four modes via the UHBI:

- Direct access to the 82750PD host I/O registers.
- Indirect I/O access to the 82750PD memory address space.
- Access to the 82750PD memory address space via an EMS-style window in the host's memory address space.
- I/O access to the SFB via a pair of Host-SFB FIFOs.

## Universal Host Bus Interface

The host I/O registers are a block of eight 32-bit registers in the host I/O space. These registers offer the programmer flexibility in configuring and controlling the other three access modes listed above.

Table 5-1 lists 82750PD register and memory categories and indicates the modes that the host can use to access them. The UHBI also supports an 82750PD interrupt to the host bus (see Section 5.7).

Table 5-1. Host Access Modes to the Multimedia System

82750PD Registers/Memory Accessible from the Host CPU	Access Modes			
	Direct I/O	Indirect I/O	"EMS" Memory Mapping*	Host-SFB FIFOs
Host I/O Registers	Yes	Yes**	No	No
Core Registers	No	Yes	Yes	No
Shared Frame Buffer (SFB)	No	Yes	Yes	Yes
Event Synchronization Registers	No	Yes	Yes	Yes

\* This is similar to true EMS mapping, but does not conform to the EMS specifications

\*\* Some 82750PD registers are accessible indirectly via the host I/O registers. The host I/O registers themselves are accessible only by direct I/O addressing

Host accesses to the multimedia system utilize the 82750PD internal bus. However, the 82750PD contains logic which ensures that the host does not monopolize the bus. For every non-82750PD core access of the internal bus, the 82750PD core is given an opportunity to access the internal bus. The following case is an exception: Under certain conditions the core can, if so programmed, allow a Host-SFB FIFO to use two consecutive internal bus cycles to access the SFB. (See the FAST bit in the read and write Host-SFB FIFOs).



The *host I/O registers* are a block of eight 32-bit registers in the host I/O space. These registers offer the programmer flexibility in configuring and controlling the other three access modes listed above.

Table 5-1 lists 82750PD register and memory categories and indicates the modes that the host can use to access them. The UHBI also supports an 82750PD interrupt to the host bus (see Section 5.7).

Table 5-1. Host Access Modes to the Multimedia System

82750PD Registers/Memory Accessible from the Host CPU	Access Modes			
	Direct I/O	Indirect I/O	"EMS" Memory Mapping*	Host-SFB FIFOs
Host I/O Registers	Yes	Yes**	No	No
Core Registers	No	Yes	Yes	No
Shared Frame Buffer (SFB)	No	Yes	Yes	Yes
Event Synchronization Registers	No	Yes	Yes	Yes

\* This is similar to true EMS mapping, but does not conform to the EMS specifications

\*\* Some 82750PD registers are accessible indirectly via the host I/O registers. The host I/O registers themselves are accessible only by direct I/O addressing

Host accesses to the multimedia system utilize the 82750PD internal bus. However, the 82750PD contains logic which ensures that the host does not monopolize the bus. For every non-82750PD core access of the internal bus, the 82750PD core is given an opportunity to access the internal bus. The following case is an exception: Under certain conditions the core can, if so programmed, allow a Host-SFB FIFO to use two consecutive internal bus cycles to access the SFB. (See the FAST bit in the read and write Host-SFB FIFOs).

**NOTE**

*All internal bus cycles to the host interface are treated as unique operations. Software must be aware that the host CPU itself may decompose a single operation into multiple bus cycles. Multiple bus cycles are not reassembled by the 82750PD. Furthermore, there is no mechanism for atomic read-modify-write operations.*

The remaining sections in this chapter describe the following UHBI topics:

- The host I/O registers and their configuration.
- Indirect access to system components via a pair of host I/O registers.
- Configuration of the EMS-style window.
- Access to the SFB via a pair of Host-SFB FIFOs.
- Configuration of the 82750PD interrupt to the host bus.
- The general configuration registers and their setup by the host.

## **5.2 Host Interface Address Configuration**

This section describes how the host addresses several devices that share the same logical "slot." For each host bus type, it describes the Programmable Option Select (POS) registers, which are used to configure the host interface addresses. By writing to the POS registers, the host specifies the location of an "I/O window" in the host I/O space and enables or disables that I/O window and the EMS-style window. For the bus types that do not employ POS registers, an equivalent set of information is given.

## 5.2.1 Configuration of Devices in a Single Slot

For the PCI and Micro Channel bus types, the devices on the SFBI appear to the host as a single logical I/O slot. This section, which applies only to the PCI and Micro Channel bus types, describes the protocol for devices sharing a single slot.

Figure 5-1 shows the I/O slots (A, B and C) and the devices for a PCI or Micro Channel bus. The devices in slot C have device IDs 0 through 7.

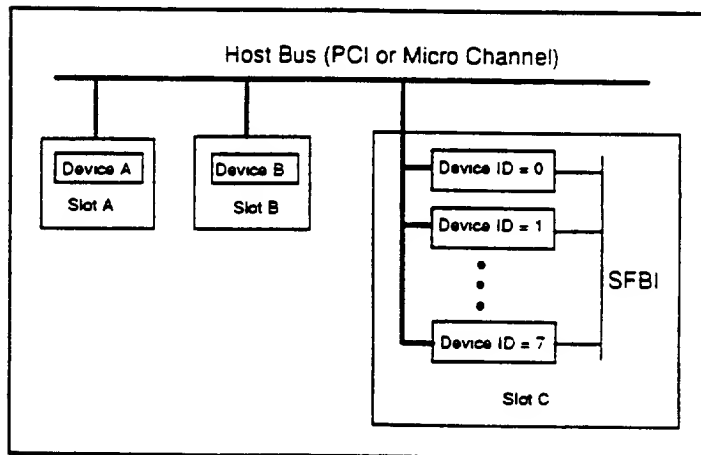


Figure 5-1. Multiple Devices in a Single Logical Slot

The following protocol is established to enable multiple devices to share a single logical slot without contention:

- Device 0 is responsible for providing all of the read data during POS (configuration) read cycles.

- Device 0 is responsible for providing all handshaking (as required for each bus) for POS read and write cycles.
- Devices 1 through 7 do not provide handshaking for POS read and write cycles.
- All devices must snoop write data and decode the bits that determine which device is to receive the data written to the slot.

The set of bits to be snooped depends on the details of the devices sharing the slot.

**NOTE**

*The 82750PD does not fully support operation as Device 0. The strap inputs should not be set for Device 0 during reset configuration.*

The 82750PD POS registers are write-only registers. The 82750PD ignores any attempt to read data from these I/O registers because the 82750PD cannot be Device 0. A different device in the system should be Device 0, and hence respond to any such read requests. The 82750PD software can read its POS register bits from the CFG registers, as described later.

### **5.2.2 Address Configuration**

Address configuration of the host interface depends on the host bus type: ISA, EISA, Micro Channel, PCI, or VL-Bus. However, for any of these host bus types the address configuration can be expressed in terms three generic fields: IO\_BASE, IO\_ENABLE, and MEM\_ENABLE. These fields are defined in Table 5-2.

Table 5-2. Generic Address Configuration Fields.

POS Field Name	Definition
IO_BASE	This 9-bit field specifies the location of the I/O window in the host I/O space
IO_ENABLE	This 1-bit field enables/disables the I/O window
MEM_ENABLE	This 1-bit field enables/disables the "EMS" window

**NOTE:**

The data values for the IO\_BASE are restricted as noted here:

**IO\_BASE Values:**

x xxxx xx00b *Legal*  
 x xxxx xx01b *Legal*  
 x xxxx xx10b *Legal*  
 x xxxx xx11b *Illegal*

The specific implementation of these generic fields depends on the bus type. The subsections that follow show the locations of the POS registers and the implementations of the three generic fields. Depending on the bus type, the values of IO\_BASE, IO\_ENABLE, and MEM\_ENABLE are permanently in a clear or set state, or determined by straps/jumpers, or programmable via the POS registers. In any case, the values of these fields can be read from the CFGINT register (MEM\_ENABLE, IO\_ENABLE, IO\_BASE[8] and the CFGBASE register (IO\_BASE[7:0], which are shown in Tables 5-36 and 5-43.

ISA POS Information

The ISA bus does not have a POS register. Table 5-3 lists the generic POS fields and their implementation in the ISA bus.

Table 5-3. ISA Implementation of POS Fields

POS Fields	ISA Implementation
IO_ENABLE	= 1 (always enabled)
MEM_ENABLE	= 1 (always enabled)
IO_BASE[8:0]	Latched from straps/jumpers at reset

EISA POS Register Format

Tables 5-4 and 5-5 show the POS registers for the EISA bus.

Table 5-4. EISA POS2 Register

Mnemonic: BCNTL

Address: 0zC84h\*

Access: W/O

Reset State: Not available.

Bit No.	7-1	0
Name	RSVD	ENABLE

Bit No.	Name	Description
0	ENABLE	ENABLE = IO_ENABLE = MEM_ENABLE. Clearing this bit disables the host I/O registers and disables the "EMS" window, even if the EWE bit (bit 0 in the CFG 06 register) is set

\* z in the Address field denotes slot-dependent bits

Table 5-5. EISA POS3 Register

Mnemonic: BASE

Address: 0zCB8Eh\*

Access: W/O

Reset State: 75h\*\*

Bit No.	0
Name	IOB[7:0]

Bit No.	Name	Description
7-0	IOB[7:0]	IOB[7:0] = I/O Base[7:0]**

\* z in the Address field denotes slot-dependent bits

\*\* If IO\_BASE[8] = 1, the reset state corresponds to IO\_START = x2EAh, where x depends on the Device ID (see Section 5.3)

\*\*\* I/O Base[8] is latched from straps or jumpers at reset

Micro Channel POS Register Format

Tables 5-6 and 5-7 show the POS registers for the Micro Channel bus.

Table 5-6. Micro Channel POS2 Register

Mnemonic: —

Address: xx02h\*

Access: W/O

Reset State: 00h

Bit No.	7-1	0
Name	RSVD	CDEN

Bit No.	Name	Description
0	CDEN	CDEN = IO_ENABLE = MEM_ENABLE. Clearing this bit disables the host I/O registers, and disables the "EMS" window, even if the EWE bit (bit 0 in the CFG 06 register) is set.

\* xx denotes eight don't-care bits

Table 5-7. Micro Channel POS3 Register

Mnemonic: —

Access: W/O

Address: xx03h\*

Reset State: 75h

Bit No.	7-0
Name	IOB[7:0]

Bit No.	Name	Description
7-0	IOB[7:0]	IOB[7:0] = IO BASE[7:0]**

\* xx denotes eight don't-care bits.

\*\* I/O Base[8] is latched from straps or jumpers at reset.

Snooping: -cdsetup must be '0' and m/-io must be '0' for POS snooping to be enabled.

PCI POS Register Format

Tables 5-8 and 5-9 show the POS registers for the PCI bus.

Table 5-8. PCI POS2 Register

Mnemonic: CMD0

Access: W/O

Address: 04h

Reset State: 00h

Bit No.	7-2	1	0
Name	RSVD	MEMEN	IOEN

Bit No.	Name	Description
0	IOEN	Clearing this bit disables the host I/O registers
1	MEMEN	Clearing this bit disables the "EMS" window even if the EWE bit (bit 0 in the CFG 06h register) is set



Table 5-9. PCI POS3 Register

Mnemonic: BASE

Address: 43h

Access: W/O

Reset State: 75h\*

Bit No.	7-0
Name	IOB[7:0]

Bit No.	Name	Description
7-0	IOB[7:0]	IOB[7:0] = I/O Base[7:0]**

\* If IO\_BASE[8] = 1 this reset state corresponds to IO\_START = x2EAh, where x depends on Device ID (see Section 5.3)

\*\* I/O Base[8] is latched from straps or jumpers at reset.

Snooping: Snooping is enabled on configuration cycles.

### VL-Bus POS Information

The VL-Bus bus does not have a POS register. Table 5-10 lists the generic POS fields and their implementation in the VL-Bus.

Table 5-10. VL-Bus Implementation of POS Fields

POS Fields	VL-Bus Implementation
IO_ENABLE	= 1 (always enabled)
MEM_ENABLE	= 1 (always enabled)
IO_BASE[8:0]	Set by 9 straps/jumpers at reset

### 5.3 Host I/O Registers

---

Eight 32-bit registers in the host I/O space are provided for the host to access the 82750PD and the SFB. The 82750PD responds when any byte in these host I/O registers is accessed by the host. The 82750PD core cannot access the host I/O registers.

Figure 5-2 shows a map of the host I/O registers. Following sections in this chapter discuss these registers in detail.

Software should not access REG 7, which is Reserved. REG7 appears in the host I/O address space and is decoded by the hardware. Bus cycles to this CFG register generate the proper acknowledgment for the bus type used and do not hang. Any data written to this CFG register is ignored. Any read of this register returns random data.

The address of the first byte of the I/O registers, REG0-BYTE0, is named IO\_START (see Figure 5-2). Table 5-11 shows how the value of IO\_START is determined. Note that IO\_START is not a physical register; it is just a label given to REG0-BYTE0.

	BYTE3	BYTE2	BYTE1	BYTE0
REG 0	PAR1 HIGH (IO_START+0003H)	PAR1 LOW (IO_START+0002H)	PAR0 HIGH (IO_START+0001H)	PAR0 LOW (IO_START+0000H)
REG 1	PAR3 HIGH (IO_START+0403H)	PAR3 LOW (IO_START+0402H)	PAR2 HIGH (IO_START+0401H)	PAR2 LOW (IO_START+0400H)
REG 2	HOST-SFB WRITE FIFO SELECTED BYTE (IO_START+0803H)	HOST-SFB WRITE FIFO CONTROL (IO_START+0802H)	HOST-SFB WRITE FIFO DATA -- HIGH (IO_START+0801H)	HOST-SFB WRITE FIFO DATA -- LOW (IO_START+0800H)
REG 3	HOST-SFB READ FIFO SELECTED BYTE (IO_START+0C03H)	HOST-SFB READ FIFO CONTROL (IO_START+0C02H)	HOST-SFB READ FIFO DATA -- HIGH (IO_START+0C01H)	HOST-SFB READ FIFO DATA -- LOW (IO_START+0C00H)
REG 4	PD INDIRECT ADDRESS EXTRA (IO_START+1003H)	PD INDIRECT ADDRESS HIGH (IO_START+1002H)	PD INDIRECT ADDRESS MID (IO_START+1001H)	PD INDIRECT ADDRESS LOW (IO_START+1000H)
REG 5	PD INDIRECT DATA BYTE3 (IO_START+1403H)	PD INDIRECT DATA BYTE2 (IO_START+1402H)	PD INDIRECT DATA BYTE1 (IO_START+1401H)	PD INDIRECT DATA BYTE0 (IO_START+1400H)
REG 6	GENERAL STATUS (IO_START+1803H)	GENERAL CONTROL (IO_START+1802H)	CFG REGISTER DATA (IO_START+1801H)	CFG REGISTER NUMBER (IO_START+1800H)
REG 7	RESERVED (IO_START+1C03H)	RESERVED (IO_START+1C02H)	RESERVED (IO_START+1C01H)	RESERVED (IO_START+1C00H)

Figure 5-2. Host I/O Registers

Table 5-11. The Value of IO\_START

Address Field	31-16	15-13	12-10	9-1	0
Contents	0	Device_ID	0	IO_BASE	RSVD

Field	Contents	Description
0	RSVD	
9-1	IO_BASE	This field is part of the POS information for each bus type
12-10	0	
15-13	Device_ID	The device ID assigned to the 82750PD
31-16	0	

A general host I/O register address has the format shown in Table 5-12. This format applies to all of the host bus types. (Note that the value of IO\_START in Figure 5-2 is the address value in Table 5-12 with REGNUM=0.)

Table 5-12. Format for the Address of a Host I/O Register

Address Field	31-16	15-13	12-10	9-1	0
Contents	0	Device_ID	REG_NUM	IO_BASE	x

Field	Contents	Description
0	x	"don't care"
9-1	IO_BASE	This field is part of the POS information for each bus type
12-10	REG_NUM	The register number (0-7) in Figure 5-2
15-13	Device_ID	The device ID assigned to the 82750PD

The following equation defines the computation of the address of a byte in a host I/O register:

$$\begin{aligned}\text{BYTE\_ADDRESS} = & \text{IO\_START} \\ & + 0400\text{H} * \text{REG\_NUM} \\ & + \text{BYTE\_NUM}\end{aligned}$$

The byte addresses in Figure 5-2 illustrate this computation.

### 5.4 .EMS-Style Memory Address Mode

---

One way the host can access the 82750PD memory space is by using an EMS-style address mode.<sup>1</sup> The "EMS" mode provides a window through the host memory address space and into the 82750PD memory address space. When the host accesses an address inside the "EMS" window, the access is redirected to an address in the 82750PD memory space. The location of the "EMS" window in the host memory address space is determined by configuration registers. Note that the host is not required to use the "EMS" mode to access the 82750PD memory address space. This space can also be accessed by other address modes (see Table 5-1). The following subsections describe the "EMS" address mode, the associated 82750PD registers, and setting up the "EMS" mode.

---

<sup>1</sup> The EMS-style address mode described here is similar to true EMS addressing but does not satisfy all of the EMS requirements. For brevity we use "EMS" to refer to this address mode.

### 5.4.1 "EMS" Memory Mapping and Registers

The "EMS" window in the host address space occupies 8 Kbytes, which are divided into four 2-Kbyte pages (see Figure 5-3). Each page of the window has an associated Page Address Register (PAR 0 - PAR 3) located in the host I/O space (REG0 and REG1). The page address register specifies a 2-Kbyte target page in the 82750PD memory address space. In this way an access to a 2-Kbyte page in the host memory address space is redirected to a 2-Kbyte page in the 82750PD memory address space.

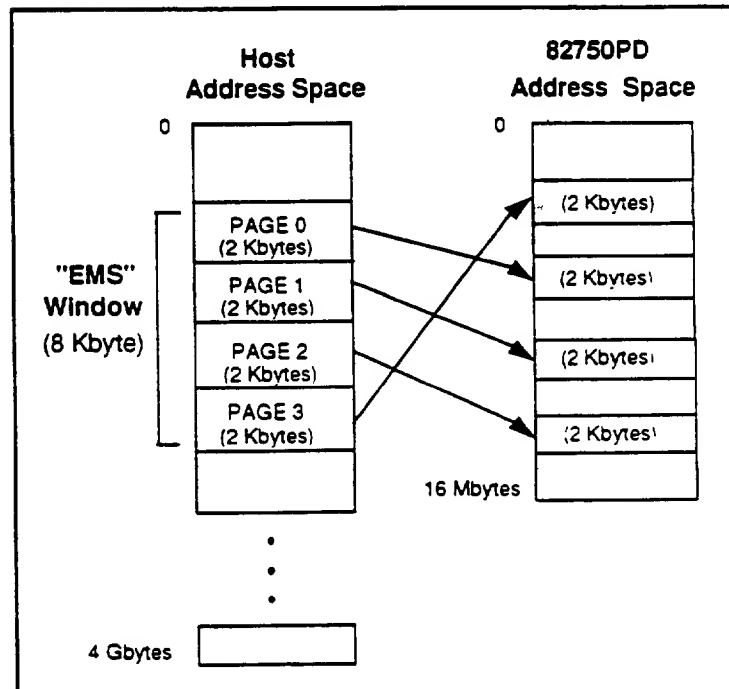


Figure 5-3. "EMS" Window Mapping

Figure 5-4 is a more detailed view of mapping a host memory address into a 82750PD memory address. When the host accesses its memory, it checks to see if the memory address is inside the "EMS" window. In Figure 5-4, the 19 bits in the Match field of the host address specify which 8-Kbyte area is about to be accessed. Three "EMS" configuration registers store 19 bits, EMS[18:0], which specify the 8-Kbyte base address of the "EMS" window. (The "EMS" configuration registers are described in Section 5.4.2.) If the bits in the MATCH field of the host memory address match EMS[18:0], then the host address is in the "EMS" window and is mapped into the 82750PD memory address space. (If there is no match, the address is used directly as an address in the host address space.)

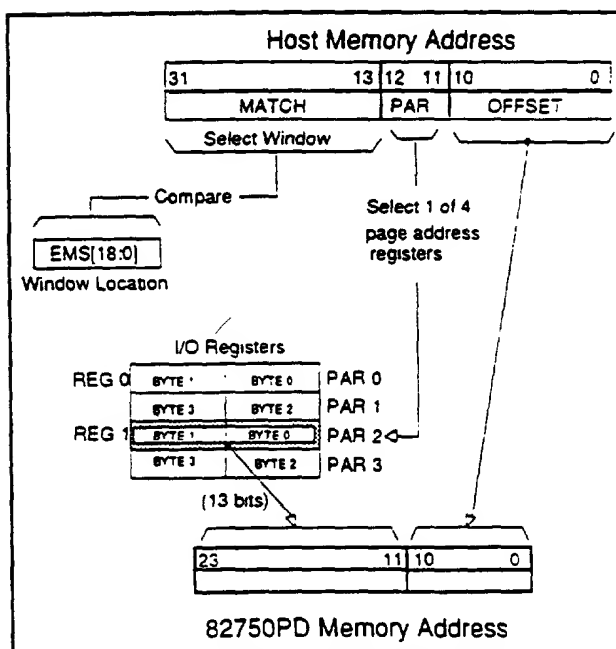


Figure 5-4. Details of "EMS" Mapping

For a host memory access that is inside the "EMS" window, the two PAR bits (bits 11 and 12) of the host address specify the page address register (0-3) that is used for the mapping. (The four PAR registers, described in Section 5.4.2, provide the programmer with up to four different mappings to use for different purposes.) The figure shows an example where the PAR bits are '1 0', which selects PAR 2. Each 16-bit page address register contains a 13-bit field that specifies the base address of a 2-Kbyte page in the 82750PD memory space. These bits (the 13 bits from PAR2 in the example) become the 13 most significant bits of the 82750PD memory address. The 11 least significant bits are taken directly from the 11 least significant bits (the Offset field) of the host memory address.

The three fields of the host memory address (OFFSET, PAR, and MATCH) can be computed from the host memory address (ADDRESS) by the following 'C' statements:

Offset Int Long;

Offset = ADDRESS & ((1 << 11) - 1);

Par\_Select = (ADDRESS >> 11) & 3;

Match = ADDRESS & !((1 << 21) - 1);

#### 5.4.2 Setting Up the "EMS" Mode

This section describes the steps to set up the "EMS" mode. These steps involve the POS registers, the "EMS" CFG registers and the Page Address Registers.



---

*POS Registers and the "EMS" Mode*

---

To enable the "EMS"-style mode, set the MEM\_ENABLE "bit" (see Section 5.2) and set the EWE bit (see below).

The implementation of the MEM\_ENABLE "bit" depends on the bus type. Information regarding MEM\_ENABLE is listed here for each bus type:

- ISA bus: MEM\_ENABLE = 1 ("EMS" is always enabled).
- EISA bus: Set the ENABLE bit in POS2.
- Micro Channel bus: Set the CDEN bit (bit 0 in POS2).
- PCI bus: Set the MEMEN bit in CMD0.
- VL-Bus: MEM\_ENABLE = 1. ("EMS" is always enabled).

---

*"EMS" Configuration Registers*

---

The "EMS" configuration registers, EMSCFG[2:0], are shown in Table 5-13. The EWE bit in EMSCFG0 must be set to enable the "EMS" mode. The EMS<sub>n</sub> bits specify the window location. Bits EMS[18:0] in the EMSCFG[2:0] registers are the 19 most significant bits of the base address of the 8-Kbyte "EMS" window. An example of setting up the EMSCFG registers follows.

Table 5-13. "EMS" Configuration Registers

Mnemonic: EMSCFG[2:0]

Address: CFG 08-06

Access: R/W

Reset State : 00h, 00h, 00h

Bit No.)	7	6	5	4	3	2	1	0
EMSCFG2	EMS18	EMS17	EMS16	EMS15	EMS14	EMS13	EMS12	EMS11
EMSCFG1	EMS10	EMS9	EMS8	EMS7	EMS6	EMS5	EMS4	EMS3
EMSCFG0	EMS2	EMS1	EMS0	RSVD				EWE

Bit No.	Name	Description
7-0	EMS[18:0]	EMS[18:0] are the 19 MSBs of the base address of the 8-Kbyte "EMS" window. The MATCH field of the host address is compared to EMS[18:0] to detect an address that is to be mapped
7-0		
7-5		
0	EWE	Setting this bit enables the "EMS" window, provided the MEM_ENABLE bit is set (see Section 5.2)

**NOTES**

1. The EWE bit and the MEMEN bit must be set to enable the "EMS" window.
2. The EWE bit must be clear while software changes EMS[18:0]. The write to EMSCFG0 that sets EWE must not change the values of EMS[2:0]. A violation of this may cause a momentary conflict that could hang the system.
3. After writing the EWE bit, software should read it back, before accessing the "EMS" window. (This ensures that the "EMS" window has had enough time to complete the configuration operation.)

As an example, assume that the "EMS" window base address is chosen to be 000D6000h. To obtain the 19 EMS bits that specify the window starting address, strip the 13 least significant bits. (or divide by 2000h) to obtain 6Bh. Place this value into bits EMS[18:0]. Doing this and setting the EWE bit gives the following values in the EMS CFG registers:

Bit No.	7	6	5	4	3	2	1	0
EMS CFG2	0	0	0	0	0	0	0	0
EMS CFG1	0	0	0	0	1	1	0	0
EMS CFG0	0	1	1	0	0	0	0	1

### Page Address Registers

Table 5-14 shows the format of the four Page Address Registers (PARs), which consist of host I/O registers REG0 and REG1. Each PAR specifies a 2-Kbyte area in 82750PD memory address space for the "EMS" mapping. Bits 2-0 of each PAR are hardwired to '0'. These bits always read as '0' and should be written as '0'. Bits 15-3 of the PAR are the 13 most significant bits of the base address, which must be on a 2-Kbyte boundary. To load a PAR register, strip the 8 least significant bits of the base address (11 bits minus the 3 hardwired bits), and write the resulting 16-bit value to the PAR register.

You must write the base address to the Page Address Register before accessing the corresponding "EMS" page.

The PARs can be written and read with either byte-I/O or word-I/O operations. You can write the bytes in any order. However, be sure to update the entire PAR register with the desired address value before accessing the corresponding "EMS" page.

Table 5-14. "EMS" Page Address Registers.

Mnemonic: PAR[3:0]

Address: REG1-BYTES3-2, REG1-BYTES1-0;  
REG0-BYTES3-2, REG0-BYTES1-0

Access: See bit descriptions.

Reset State: 0000h

Bit No.	15-3	2	1	0
Name	PAR	0	0	0

Bit No.	Name	Description
3-0	0	These bits are wired to '0'. They read as '0' and should be written as '0'.
15-3	PAR	These R/W bits are the 13 MSBs of the base address of a 2 Kbyte page in the 82750PD memory address space. A host access to page <i>n</i> of the "EMS" window is redirected to this page (provided the "EMS" mode is enabled).

## 5.5 82750PD Indirect Access

The 82750PD Indirect Access mode provides a way for the host to indirectly access the 82750PD internal bus memory space through two 32-bit registers in the host's I/O space. Figure 5-5 illustrates this access mode.

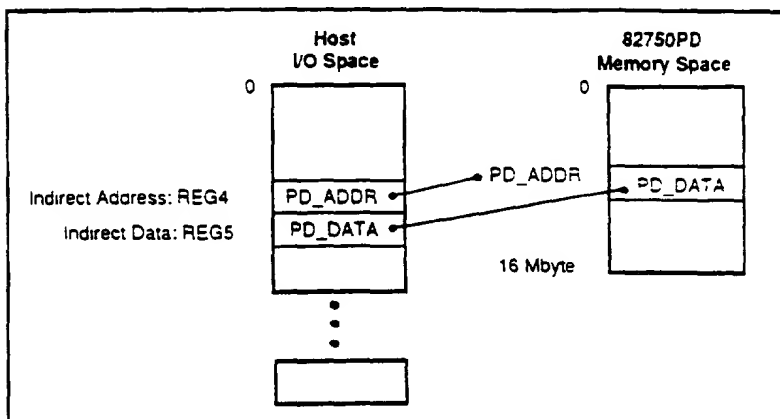


Figure 5-5. 82750PD Indirect Register Mapping

The host's indirect address register (see Table 5-15) holds the address of the dword in the 82750PD memory space that is accessed (PD\_ADDR). The host's indirect data register (see Table 5-16) holds the data (PD\_DATA) that is written to (or read from) that address. When the host accesses the 82750PD indirect data register, the 82750PD generates an internal bus read or write cycle to access the dword in the 82750PD memory space.

Table 5-15. Indirect Address Register

Mnemonic: — Address: REG4-BYTES3—REG4-BYTE0  
Access: R/W Reset State: 00000000h

Bit No.	31-0
Name	—

Bit No.	Name	Description
15-0	—	These four bytes are the 82750PD memory address for an indirect access of a location in the 82750PD memory address space

Table 5-16. Indirect Data Register

Mnemonic: — Address: REG5-BYTE3—REG5-BYTE0  
 Access: R/W Reset State: Uninitialized

Bit No.	31—0
Name	—

Bit No.	Name	Description
15—0	—	These four data bytes are data read from or written to the 82750PD memory for an indirect access of the 82750PD memory address space. The 82750PD address for the indirect access is in REG4-BYTE3—REG4-BYTE0.

To indirectly access a location in the 82750PD memory address space, execute the following two steps :

1. Write the 82750PD address to be accessed (PD\_ADDR in Figure 5-5) to the host's Indirect Address register (REG4-BYTE3—REG4-BYTE0).
2. Read/Write the 82750PD data (PD\_DATA in Figure 5-5) from/to the 82750PD Indirect Data register (REG5-BYTE3—REG5-BYTE0).

The indirect access can be a byte-I/O, word-I/O, or dword-I/O operation. The bytes accessed in the 82750PD memory dword are the same bytes as those accessed in REG5. The order of accessing the bytes in the 82750PD memory dword is the same as the order of accessing the bytes in REG5.

You may access the bytes of REG5 in any order; however you must update the entire 82750PD indirect address register (REG4) with the desired address value before accessing the 82750PD indirect data register.

An access to REG5 with IO\_BASE[0] = 1 activates internal byte swapping logic in the 82750PD. For an

example with IO\_BASE = 75h and 82750PD device ID = 1, a computation as specified in Section 5.3 yields the following addresses for the bytes of REG5:

REG5-BYTE0 = 34EAh  
REG5-BYTE1 = 34EBh  
REG5-BYTE2 = 34ECh  
REG5-BYTE3 = 34EDh

For example, when a dword I/O write instruction is executed, the host splits this into two host write cycles because this is a non-aligned operation. For each of these host cycles, the data bytes are on the wrong 16-bit lines to go directly to the specified bytes of REG5. Hence, for this case the 82750PD internally swaps these data lines in order to reference the correct bytes of REG5. Each of these host cycles causes a separate write to memory with appropriate byte enables set.

The indirect data register is "pass-through" (there is no buffering). Host cycles access the 82750PD memory space directly. The 82750PD indirect data register is uninitialized at reset because no physical indirect data register exists.

Figure 5-6 shows the generation of the 82750PD address from the host I/O address and the address held in the 82750PD indirect address register (REG4). The least significant two bits and the most significant eight bits of the 82750PD indirect address register are never used. Accordingly, these bits are hardwired to '0'. Any value written to these bits is ignored. Software should always write these bits as '0'.

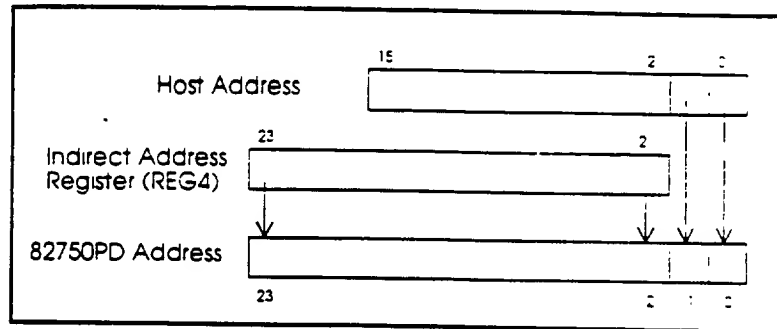


Figure 5-6. 82750PD Indirect Register Address Translation

## 5.6 Host-SFB FIFOs

The Host-SFB FIFOs, shown in Figure 5-7, provide a high performance data path for the host to read or write the SFB. The host accesses the FIFOs in the host's I/O address space. In a single data transaction the FIFO reads or writes one dword. The FIFOs are useful for accessing the SFB in high performance modes or through host DMA channels.

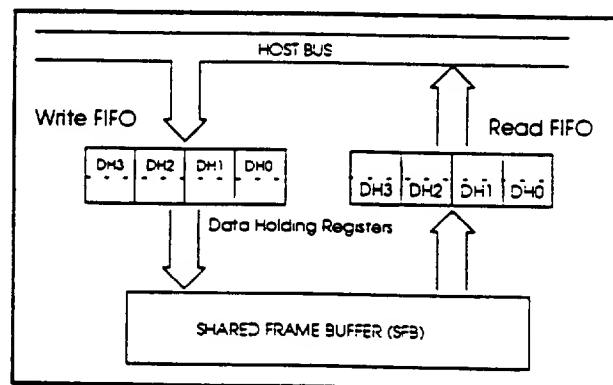


Figure 5-7. Block Diagram of the Read and Write FIFOs



The FIFO data holding registers (DH3-DH0 in Figure 5-7) are four bytes wide. The double buffered design allows the host and the SFB to access the data holding register simultaneously without contention. This design maximizes the throughput of the host bus and the 82750PD internal bus.

Both FIFOs can be programmed to operate in an auto-increment mode, which causes the SFB address to increment by four after each FIFO access.

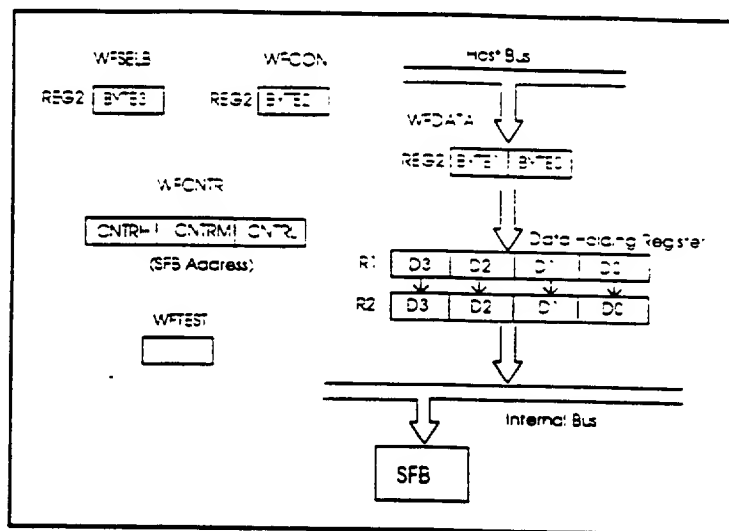
### NOTE

*The FIFOs contain no consistency checking hardware. It is software's responsibility to handle any "stale" data problems (typically by re-initializing or flushing the FIFOs).*

### 5.6.1 Host-SFB Write FIFO

---

The Host-SFB Write FIFO, shown in Figure 5-8, is a high-speed data path from the host to the SFB. The data to be written to the SFB is placed in the Write FIFO Data Register (WFDATA). The Write FIFO Address Counter (WFCNTR) consists of bytes CNTRH, CNTRM, and CNTRL. The data in WFDATA is written to the 24-bit address held in WFCNTR. Registers R1 and R2 comprise the double-buffered data holding register, which holds the dword of data to be written to the SFB. Write FIFO operations are controlled by the Write FIFO Control Register (WFCN). The Write FIFO Test Register (WFTST) provides diagnostic information.



**Figure 5-8. Block Diagram of the Write FIFO**

The host communicates with the Write FIFO via one 32-bit I/O register, REG2. Table 5-17 lists the addresses and names of the Write FIFO registers. The WFCNTR register bytes (CNTRH, CNTRM, CNTRL) and the WFTEST register are accessed indirectly through the Write FIFO Selected Byte Register (WFSELB).

**Table 5-17. Host-SFB Write FIFO Registers**

Address	Mnemonic	Register Name
REG2-BYTE1—REG2-BYTE0	WFDATA	Write FIFO Data Register
REG2-BYTE2	WFCON	Write FIFO Control Register
REG2-BYTE3	WFSELB	Write FIFO Selected Byte Register
	WFCNTR*	Write FIFO Address Counter
	WFTST*	Write FIFO Test Register

\*These registers are accessed via WFSELB

### Basic Operation of the Host-SFB Write FIFO

This section briefly describes the operation of the Write FIFO. Details are given in following sections on the registers and on specific procedures.

To write to memory via the Write FIFO, the host first writes the three bytes of the starting SFB address to the three bytes of the WFCNTR register. The host then writes data to the WFDATA register. After each write, the hardware moves the data to the data holding register, and the FIFO uses a normal internal bus cycle to move the data from the data holding register to the SFB.

Two items to be discussed later should be noted here:

- The FIFO waits until the host writes to the most significant byte (DH3) of the data holding register before writing the data to the SFB. This means that if DH3 is not loaded when the last data is written to the WFDATA register, software must command the last data to be written to the SFB.
- There is no byte-alignment hardware between the WFDATA register and the data holding register. As a result the software must attend to loading data bytes into the WFDATA register. (See "Loading the Write FIFO Data Register.")

The 32-bit data holding register (R1 and R2 in Figure 5-8) is double-buffered to increase the throughput of the host bus and the 82750PD internal bus. While the FIFO is delivering the dword in R2 to the SFB, the host can write a new dword to register R1. (The data transfer from R1 to R2 is transparent to the programmer, who treats R1 and R2 as a single register.)

Write FIFO Control Register (WFCON)

The Write FIFO Control Register (REG2-BYTE2) is shown in Table 5-18. Following paragraphs discuss the bits. (Full bit descriptions are given in Appendix A.)

**NOTE:**

*Before altering the WFCON register, software should verify that the Write FIFO is empty (see the EMPTY bit in the WFCON register).*

Table 5-18. Write FIFO Control Register

Mnemonic: WFCON

Address: REG2-BYTE2

Access: R/W

Reset State: 08h

Bit No.	7	6	5	4	3	2	1	0
Name	AUTO	TEST	R:FULL* W:RSVD	FAST	R:EMPTY* W:RSVD	BS2	BS1	BS0

\* R = Read, W = Write

**BS2-BS0** (Byte Select, bits 2-0). The Byte Select field, comprising bits BS2, BS1 and BS0, selects the bytes or operations listed in Table 5-19. The values of BS in the table represent the combinations of the BS2-BS0 bits. The bytes in Table 5-19 are accessed by reading or writing the WFSELB register.

Table 5-19. Byte Select Bits in the WFCON Register

BS2	BS1	BS0	BS	Write	Read
0	0	0	0	CNTRL	CNTRL
0	0	1	1	CNTRM	CNTRM
0	1	0	2	CNTRH	CNTRH
0	1	1	3	FLUSH*	TEST*
1	0	0	4	CNTRL	DH0
1	0	1	5	CNTRM	DH1
1	1	0	6	CNTRH	DH2
1	1	1	7	FLUSH*	DH3

\* FLUSH denotes an operation

The BS values 0, 1, and 2 select bytes of the WFCNTR register, which holds the starting address of the next dword to be written to the SFB (or read from the SFB in TEST mode). (The BS values 4, 5, and 6 serve the same purpose for writes only.) A following section, "Write FIFO Address Counter (WFCNTR)," gives a procedure for writing the SFB address to this register.

Writing BS = 3 or 7 to the control register "flushes" the data holding register. A more detailed look at the hardware in Figure 5-8 is helpful here. Whenever the FIFO hardware transfers data from R1 to R2, the FIFO tries to write the data to the SFB (it requests the internal bus). Data is not moved from R1 to R2 until new data is written from the data register to the most significant byte (DH3) of R1. This means that R1 could have data in any or all of the lower bytes (DH0-DH2), and the data would not be moved to R2 as long as DH3 is not written.

The flush operation provides a way to move data from the DH0-DH2 bytes of the data holding register to the SFB without writing to DH3. Upon execution of a flush, the FIFO waits, if necessary, for the data in R2 to be written to the SFB, and then moves data in R1 to R2, from where it is written to the SFB.

nothing is written to the corresponding bytes in memory when the flush is executed. After a flush, the FIFO is "empty" (see the EMPTY bit). If there is data in the data holding register that would finally be moved to the SFB (in normal operation), and if software executes a flush, the final result is the same as if the flush were not executed. In other words, a superfluous flush does not change the result.

Reading the WFSELB register with BS = 3 reads the WFTST register. This register, which provides information regarding the data holding registers, is described in the section "Write FIFO Test Register."

Reading the WFSELB register with BS = 4-7 reads the data holding register bytes DH0-DH3. At reset, the data holding bytes are uninitialized.

**EMPTY** (Empty, bit 3). When the control register is read, the EMPTY bit (bit 3) reflects the status of the data holding registers R1 and R2. If the host has not written to R1 since the FIFO has written the contents of both R2 and R1 to the SFB, the FIFO is "empty," and the EMPTY bit is set. The EMPTY bit is also set following a flush. When writing to the control register, write a '0' to bit 3. Reading bit 3 always returns the value of the EMPTY bit, regardless of the value written to that bit.

Software must verify that the FIFO is "empty" before it alters the WFCNTR register or WFCON register. If these registers are altered when EMPTY = 0, a pending data operation can fail. Such alterations include changes to the AUTO bit, the TEST bit, or the TCLK bit.

**FAST** (Fast, bit 4). Under certain conditions, setting the FAST bit allows the FIFO to write two dwords from the data holding register (R1 and R2) to the

SFB in two consecutive internal bus cycles. The following conditions are required:

- FULL = 1 (the FIFO is full).
- WFCNTR points to an even dword address (an address whose least significant bits are '00').
- AUTO = 1 (the FIFO is in auto-increment mode).
- FAST = 1.

This increases efficiency by saving the overhead of arbitration and transfer of control for writing the second dword. For FAST = 0, the 82750PD executes normal bus cycles.

**FULL (R)**, (Full, bit 5). For a control register read, the FULL bit reflects the status of the holding registers R1 and R2. The FULL bit is set only if both of the following conditions hold:

- Register R2 holds valid data, and the FIFO has not yet written that data to the SFB.
- The host has written data to (at least) the most significant byte of register R1.

Since the FIFO requests an internal bus cycle when (at least) the most significant byte of R1 is valid, it is entirely possible that the FIFO is never full in normal operation.

If the FIFO is full, then setting the FAST bit speeds the writes to the SFB under certain conditions (see the FAST bit). If the FIFO is full and the host writes to the WFDATA register, wait states are inserted in the host bus cycle until the FIFO can accept the data from the host.

(Note that having the FULL bit set is not the complement of having the EMPTY bit clear. For

example, if DH1 is the only holding register byte with valid data, the FIFO is neither full nor empty.

**TEST** (Test, bit 6). Setting this bit prevents the internal bus acquisition logic from requesting the bus. This allows the host to access the Write FIFO registers without triggering a write cycle. This is useful in the following two situations:

- FIFO initialization. The TEST bit must be set during initialization of the FIFO.
- FIFO diagnostics. Inhibiting the write cycle facilitates debugging.

Note that setting the TEST bit does not interrupt an internal bus cycle that is already in process, i.e., once the FIFO has requested the internal bus, the write eventually occurs. If TEST = 1 and the FIFO is full, a write to the WFDATA register is ignored.

For diagnostic purposes, the TEST bit may stay set through several register reads and writes. Previous FIFO implementations required software to set the test bit at the beginning of each operation to ensure correct initialization (prevent a false cycle). The 82750PD implementation does not require the TEST bit to be set repeatedly; however, it does run compatibly with software that does set the TEST bit before each operation.

**AUTO** (Automatic Increment, bit 7). If the AUTO bit is set, the WFCNTR register increments by four after each write data cycle on the internal bus. If the FIFO writes two dwords in consecutive internal bus cycles (see the discussion of the FAST bit), it increments by 8. (Actually, there are two increments by 4. The difference is moot except under unusual circumstances, such as in a diagnostic program, where it might be possible for the software to catch the intermediate value.) If



ALTO = 0, the Write FIFO accesses the same location repeatedly until a new address is written.

### Write FIFO Address Counter (WFCNTR)

The three bytes (CNTRH, CNTRM, CNTRL) of the WFCNTR register hold the starting address of the next dword to be written to the SFB. The WFCNTR register is cleared at reset. To write the SFB address to the WFCNTR register, execute the sequence of writes indicated in Table 5-20. Accesses to the three bytes of the WFCNTR register must be in the order shown. You can write to the WFCNTR register (REG2-BYTE2) and the WFCNTR register (REG2-BYTE3) in the same instruction.

#### **NOTE:**

*Before altering the WFCNTR register, software should verify that the Write FIFO is empty (see the EMPTY bit in the WFCNTR register).*

Table 5-20. I/O Sequence to Read/Write the SFB Address

Step	Value of BS in WFCNTR Register (REG2-BYTE2)	Byte of WFCNTR Accessed the WFCNTR Register (REG2-BYTE3)
1	BS = 0	CNTRL (low byte)
2	BS = 1	CNTRM (middle byte)
3	BS = 2	CNTRH (high byte)

The host-to-SFB data transfer can start and end on any byte boundary in the SFB, and data can be moved from any byte boundary in the host memory. However, some rules must be followed to obtain the desired byte alignments in the SFB. (See "Loading the Write FIFO Data Register.")

---

*Write FIFO Data Register (WFDATA)*

---

The Write FIFO Data Register (REG2-BYTE1—REG2-BYTE0) is a write-only register. (Reading this register returns undefined data, and does not affect the Write FIFO). The WFDATA register is uninitialized at reset.

The WFDATA register can be written 8 or 16 bits at a time. However, when writing to the data register, observe these rules:

- When writing with byte operations, always write both bytes of WFDATA and alternate the writes to BYTE1 and BYTE0.
- Do not access the WFDATA register and the WFCON register in the same host bus cycle. Doing so may leave false data in the data register.

---

*Loading the Write FIFO Data Register*

---

The diagram in Figure 5-9 shows the data paths for bytes moving from the WFDATA register to the data holding register and, finally, to the SFB. Note that data transfers from the Write FIFO (from the data holding register) to the SFB are in dword units, and the dword is stored on a dword boundary. In addition, transfers from the WFDATA register to the holding register are in word units. To store bytes in the desired SFB locations, software must compensate for the lack of byte-alignment hardware. This section gives a data-loading procedure that results in the data being stored in the correct SFB address.

A brief look at the Write FIFO hardware points to the procedure for loading the WFDATA register. The diagram in Figure 5-9 shows the data paths for

bytes moving from the WFDATA register, to the data holding register, and to the SFB.

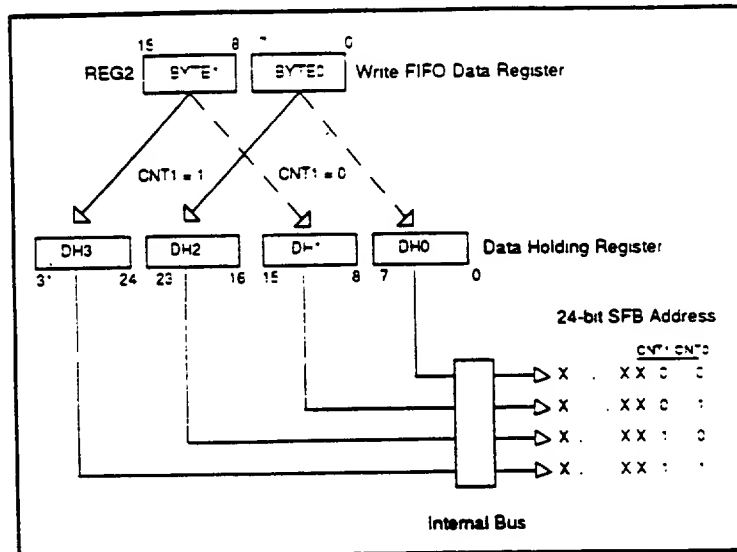


Figure 5-9. Data Paths from the Write FIFO Data Register to the SFB

The hardware examines bit 1 (CNT1) of the SFB address, (i.e., bit 1 of the least significant byte (CNTRL) in the WFCNTR register). The hardware then moves the word in the 16-bit data register to the 32-bit data holding register according to the following rules:

- If CNT1 = 0, BYTE1-BYTE0 is moved to bytes DH1-DH0 of the holding register (dashed arrows in Figure 5-9).
- If CNT1 = 1, BYTE1-BYTE0 is moved to bytes DH3-DH2 of the holding register (solid arrows in Figure 5-9).

To coordinate with the hardware, the software should follow this procedure:

1. Write the SFB (destination) address of the first data byte to the WFCNTR register.
2. Load data into the WFDATA register according to these directions:
  - If the SFB address of the first byte is odd (CNT0 = 1), move the first data **byte** to REG2-BYTE1 of the data register. (For CNT0 = 1, the data in BYTE0 is a "don't care"; it is ignored by the hardware.)
  - If the SFB address of the first byte is even (CNT0 = 0), move the first data **word** to the data register, and then move the second word to the data register.

(The hardware moves each word to DH1-DH0 and DH3-DH2 according to bit CNT1, as described above.)

3. Continue moving data words to the WFDATA register.

To ensure correct results, be careful with the I/O address, the I/O transfer size, and the WFCNTR register. The two least significant bits (CNT1, CNT0) of the WFCNTR register always correctly reflect the SFB address of the next byte to be written. Programming examples illustrating how to begin a data move from the host to the SFB are in Appendix C.

#### Saving and Restoring the State of the Write FIFO

The following is a procedure for saving the state of the Write FIFO:

1. Read the WFCON register and save it for restoring the FIFO.

## Universal Host Bus Interface

2. If the EMPTY bit is set, go to step 4.
3. Flush the FIFO and wait until the EMPTY bit in the WFCNTR register is set.
4. Read and save the 24-bit WFCNTR register. (This is the SFB address of the next byte to be moved.)
5. Restore the control register state that was saved in step 1.

The FIFO can now be initialized and used for another transfer.

"Restoring" the state of the FIFO is indistinguishable from beginning a new operation. Write the SFB starting address for the next data transfer to the WFCNTR register and proceed with the new operation.

### Write FIFO Test Register

The Write FIFO Test Register (WFTEST) provides diagnostic information concerning the data holding registers. To read the WFTEST register, read the WFSELB register with BS = 3. The bits of this register (see Table 5-21) indicate whether the bytes in the data holding registers are empty, i.e., not written since: (1) initialization, or (2) a flush, or (3) the latest transfer from R2 to the SFB.

Table 5-21. Write FIFO Test Register

Mnemonic: WFTTEST

Address: REG2-BYTE3 with BS = 3

Access: R/O

Reset State: 0FFh

Bit No.	7-4	3-0
Name	R1DH[7:4]	R2DH[3:0]

Bit No.	Name	Description
3-0	R2DH $n$	The R2DH $n$ bit ( $n = 0, 1, 2, 3$ ) of data holding register R2 is set if bit $n$ of data holding register R2 is empty.
7-4	R1DH $n$	The R1DH $n$ bit ( $n = 0, 1, 2, 3$ ) of data holding register R1 is set if bit $n$ of data holding register R2 is empty.

### 5.6.2 Host-SFB Read FIFO

The 32-bit Read FIFO, shown in Figure 5-10, is a high-speed data path from the SFB to the host. The host accesses the Read FIFO via one 32-bit I/O register, REG3. Two bytes (REG3-BYTE1—REG3-BYTE0) comprise the Read FIFO Data Register (RFDATA). REG3-BYTE2 is the Read FIFO Control Register (RFCON), and REG3-BYTE3 is the Read FIFO Selected Byte Register (RFSELB). The RFSELB register is used to access the three bytes (CNTRL, CNTRM, and CNTRH) of the Read FIFO Address Counter Register (RFCNTR). (The RFSELB register is also used to access the Read FIFO Test Register, which is described later.) Registers R1 and R2 comprise the double-buffered data holding register. While the host reads the data in R2 via the RFDATA register, the FIFO can fetch the next dword of data from the SFB and place it into R1. This double-buffered design maximizes the throughput of the host bus and the 82750PD internal bus. Table 5-22 lists the Host-SFB Read FIFO registers.

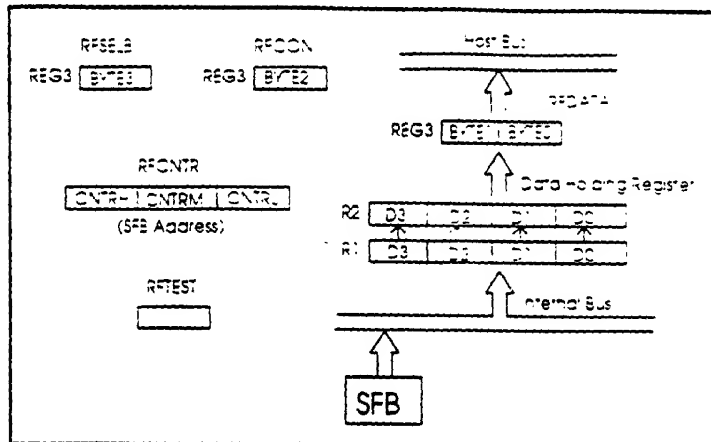


Figure 5-10. Block Diagram of the Read FIFO

Table 5-22. Read FIFO Registers

Address	Mnemonic	Register Name
REG3-BYTE1, REG3-BYTE0	RFDATA	Read FIFO Data Register
REG3-BYTE2	RFCON	Read FIFO Control Register
REG3-BYTE3	RFSELB	Read FIFO Selected Byte Register
	RFCNTR*	Read FIFO Address Counter
	RFTEST*	Read FIFO Test Register

\* These registers are accessed via the RFSELB Register

### Basic Operation of the Host-SFB Read FIFO

This section describes the basic operation of the Read FIFO. (The Read FIFO operates differently according to whether the host is reading the data in (i) single bytes or words or (ii) blocks. These differences are described in the following section on the control register (RFCON)).

To read the SFB memory via the Read FIFO, the host first loads the RFCNTR register with the starting SFB address. This requires three write operations. Each write simultaneously assigns the RFSELB register to one byte of the RFCNTR register and writes one byte of the SFB address to the RFSELB register. The three SFB address bytes must be written in order from the lowest byte to the highest byte: CNTRL, CNTRM, CNTRH. The write to CNTRLH signals the Read FIFO to fetch data from the SFB.

The Read FIFO fetches one dword (never a single word or byte) and places it into the input buffer (R1) of the data holding register. The next step depends on whether buffer R2 has data that is yet to be read by the host (via the RFDATA register). If R2 contains data, the FIFO waits until the host has read that data and then moves the new dword from R1 to R2. If register R2 is empty, the new dword is moved immediately from R1 to R2. Then, while the host is reading the dword in R2, the Read FIFO fetches the next dword and places it in R1. This prefetch ensures that the host can read the next dword without waiting for an internal bus cycle.

Note that while the FIFO fetches data in dwords, the host reads the data from the RFDATA register in words or bytes. The section, "Reading the Read FIFO Data Register," describes how the host software can accommodate this difference.

---

#### *Read FIFO Control Register (RFCON)*

---

Table 5-23 shows the RFCON register. Paragraphs following the table discuss the operation of the Read FIFO in terms of the state of the RFCON register. (Full bit descriptions are given in Appendix B.)



Table 5-23. Read FIFO Control Register

Mnemonic: RFCON

Address: REG3-BYTE2

Access: R/W

Reset State: 00h

Bit No.	7	6	5	4	3	2	1	0
Name	AUTO	TEST	RSVD	FAST	R EMPTY W: RSVD	BS2	BS1	BS0

\* R = Read, W = Write

**BS2-BS0** (Byte Select, bits 2-0). The Byte Select field, comprising bits BS2, BS1 and BS0, is used to select a byte in the RFCNTR register (for both reads and writes) or to select the Read FIFO Test Register (RFTEST) for reads. Table 5-24 shows these selections. The values of BS in the table represent the combinations of the BS1-BS0 bits. Note that the BS2 bit is a "don't care." The RFTEST register is described in a section below.

Table 5-24. Byte Select Bits in the RFCON Register

BS2	BS1	BS0	BS	Write	Read
X	0	0	0	CNTRL	CNTRL
X	0	1	1	CNTRM	CNTRM
X	1	0	2	CNTRH	CNTRH
X	1	1	3	NOP	RFTEST

**EMPTY** (Empty, bit 3, R/O). The EMPTY bit reflects the status of the data holding registers, R1 and R2. The EMPTY bit is set if neither R1 nor R2 contains data. In AUTO mode, the EMPTY bit would normally be set only for a short time after the SFB address has been loaded. The EMPTY bit and the FULL bit are not complements.

If the FIFO is empty, then setting the FAST bit speeds the reads from the SFB under certain conditions (see the FAST bit). If the host reads the RFDATA register when the EMPTY bit is set, wait

states are inserted in the host bus cycle until the read is finished.

**FAST** (Fast, bit 4). Under certain conditions, setting the FAST bit allows the FIFO to read two dwords from the SFB and place them in the data holding register (R1 and R2) in two consecutive internal bus cycles. The following conditions are required:

- **EMPTY** = 1 (the FIFO is full).
- **WFCNTR** points to an even dword address (an address whose least significant bits are '00').
- **AUTO** = 1 (the FIFO is in auto-increment mode).
- **FAST** = 1.

This increases efficiency by saving the overhead of arbitration and transfer of control that would normally be incurred in reading the second dword. For **FAST** = 0, the 82750PD executes normal bus cycles.

**FULL** (R/O) (Full, bit 5). When the host reads the **RFCNTR** register, bit 5 is the **FULL** bit. This bit reflects the status of the data holding registers, R1 and R2. The **FULL** bit is set if register R1 has data and the high byte (DH3) of register R2 has data, i.e., has not been read.

Software should verify that the FIFO is "full" before it alters the **RFCNTR** or **RFCNTR** register. (The full state is stable.) Altering these registers when the FIFO is not full can cause a pending data operation to fail. Such alterations include changes to the **AUTO** bit, the **TEST** bit, and the **TCLK** bit and a change of address in **RFCNTR**.

If the **FULL** bit is not set, the Read FIFO contains some residual data because the Read FIFO tries to fill the data holding registers whenever the most

significant byte of a 32-bit word is read by the host. If the FIFO is not in the AUTO mode, and if a transfer to the host ends on the first, second, or third byte (DH0, DH1, or DH2) then the FIFO contains some residual data. When in the AUTO mode, the Read FIFO always contains some residual data after the transfer is finished. This unwanted data is a by-product of reading ahead and trying to keep the FIFO full. Since the FIFO controller always attempts to keep up with the host reads, the normal operating condition of the FIFO is "full." To purge residual data, take the FIFO out of AUTO mode and read (at least) byte DH3 of R2.

**TEST** (Test, bit 6). The TEST bit is used for Read FIFO diagnostics. Setting this bit initializes the FIFO and prevents the internal bus acquisition logic from requesting the bus. This allows the host to access the Read FIFO registers without triggering a read cycle. Note that setting the TEST bit does not interrupt an internal bus cycle that has already begun: once the FIFO has requested the bus, the read eventually occurs. Reading an empty Read FIFO with TEST = 1 returns erroneous data. Furthermore, IOCHRDY is never asserted (the read cycle is never initiated).

For diagnostic purposes, the TEST bit may stay set through several register reads and writes. Previous FIFO implementations required software to set the test bit at the beginning of each operation to ensure correct initialization (prevent a false cycle). The 82750PD implementation does not require software to set the TEST bit repeatedly; however, it runs compatibly with software that does set the TEST bit before each operation.

**AUTO** (Auto-increment, bit 7). Setting the AUTO bit puts the Read FIFO in auto-increment mode, which facilitates reading a block of data from the SFB. To read a single dword, word, or byte, the AUTO bit

should be clear. When the AUTO bit is set, the RFCNTR register increments by four after each read cycle of the internal bus, thus setting up the address counter for reading the next dword. When the Auto bit is clear, the RFCNTR register does not increment automatically. The Read FIFO accesses the same SFB address repeatedly until a new address is written to the RFCNTR register.

When the AUTO bit is set, the Read FIFO fetches a dword as soon as the host writes the high byte of the RFCNTR register. It continues to prefetch dwords as long as the data deposited into R1 can be moved immediately into R2 (as a result of the host reading the RFDATA register). However, if the FIFO becomes empty (EMPTY = 1), the Read FIFO fetches two dwords from the SFB in consecutive internal bus cycles, provided other conditions are met (see the FAST bit).

When the AUTO bit is clear, the Read FIFO does not prefetch data. It fetches data only when the host attempts to read the RFDATA register and there is no data in R2. The fetch is then a Slow SFB cycle: wait states are inserted into the host bus cycle until the RFDATA register receives data.

#### Read FIFO Address Counter (RFCNTR)

The three bytes (CNTRH, CNTRM, CNTRL) of the Read FIFO Address Counter (RFCNTR) hold the address of the next dword to be read from the SFB. Accessing the 24-bit address involves both the RFCON register (REG3-BYTE2) and the RFSELB register (REG3-BYTE3). To access the RFCNTR register, execute 16-bit I/O operations in the order shown in Table 5-25. In auto-increment mode (AUTO = 1) the FIFO fetches data from the SFB when the high byte of the address is written to the CNTRH byte.

Table 5-25. I/O Sequence for RFCNTR Register Accesses

Step	Value of BS in the RFCNTR Register (REG3-BYTE2)	Byte of RFCNTR Accessed through the RFSELB Register (REG3-BYTE3)
1	BS = 0	CNTRL (low byte)
2	BS = 1	CNTRM (middle byte)
3	BS = 2	CNTRH (high byte)

The SFB-to-host data transfer can start and end on any byte boundaries in the SFB. However, some rules should be followed to obtain the desired byte alignments. (See the section "Reading the Read FIFO Data Register.")

#### Read FIFO Data Register (RFDATA)

The Read FIFO Data Register (RFDATA at REG3-BYTE1) accepts data from the SFB (via the data holding register) and holds it for reading by the host. At reset this register is uninitialized. The RFDATA register is read-only. (If the register is written, the write cycle completes, but the Read FIFO is unchanged.) The RFDATA register can be read with word or byte operations. However, the following rules should be observed:

- When reading the RFDATA register with byte operations, read the high byte (REG3-BYTE1) between any two reads of the low byte (REG3-BYTE0).
- Do not access the RFDATA register and the RFCNTR register in the same host bus cycle.

The RFDATA register is not initialized at reset.

There is no byte alignment hardware between the data holding register and the RFDATA register. The section "Reading the Read FIFO Data Register"

describes steps reading the desired data from the SFB.

### Read FIFO Test Register (RFTEST)

The Read FIFO Test Register (RFTEST) provides the programmer with diagnostic information. This register is accessed by reading the RCON register with BS = 3 or 7. This register should be written as 00h, although the value written is ignored. Table 5-26 describes the RFTEST register.

Table 5-26. Read FIFO Test Register

Mnemonic: RFTEST

Address: REG3-BYTE3, BS = 3 or 7

Access: R/O

Reset State: 67h

Bit No.	7	6	5	4	3	2	1	0
Name	REQ	SEL	RFRES	FAST	CNT2	NXTFAST	HR1	HR2

**HR2** (Holding Register 2, bit 0). If this bit is set, data holding register R2 (host side) contains valid data.

**HR1** (Holding Register 1, bit 1). If this bit is set, data holding register R1 (SFB side) contains valid data.

**NXTFAST** (Next Fast, bit 2). If this bit is set, the Read FIFO is requesting a 64-bit internal bus cycle.

**CNT2** (CNT2, bit 3). This bit is bit 2 of the RCON register.

**FAST** (Fast, bit 4). This is the FAST bit (bit 4) of the RCON register.

**RFRES** (Read FIFO Reset, bit 5). If this bit is set, the Read FIFO is being reset. The FIFO enters its reset state upon reset of the 82750PD or when the

## Universal Host Bus Interface

host writes the low byte of the RFCNTR register. The FIFO exits its reset state when the host writes the high byte of the RFCNTR register.

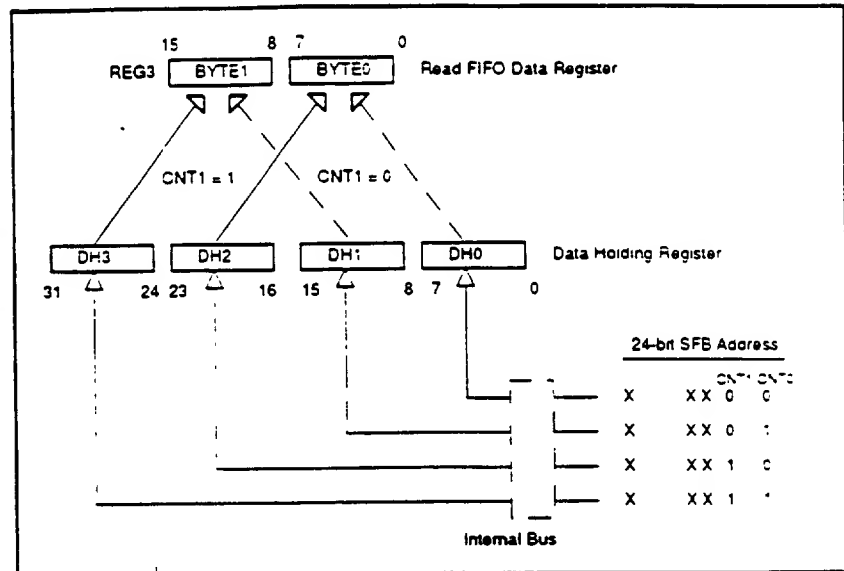
**SEL** (Select, bit 6). If this bit is clear, the internal bus arbiter has selected the Read FIFO.

**REQ** (Request, bit 7). If this bit is set, the Read FIFO is requesting use of the 82750PD internal bus.

## Reading the RFDATA Register

The diagram in Figure 5-11 shows the data paths for bytes moving from the SFB to the data holding register and then to the RFDATA register. Data transfers from the SFB to the data holding register are in dword units beginning at a dword boundary in the SFB. Data transfers from the data holding register to the RFDATA register are in word units. Hardware moves the high or low word from the data holding register to the WFDATA register according to the value of bit 1 (CNT1) of the address.

If a word or a byte is read from an even address, hardware loads the that word (or the word containing that byte) into the RFDATA register with the low byte in BYTE0. In this case no software adjustment is needed.



**Figure 5-11. Data Paths from the SFB to the Read FIFO Data Register**

If a word or a byte is read from an odd address, hardware loads that word (or the word containing that byte) into the RFDATA register with the low byte in BYTE1. For the case of a single byte read, software must take that byte from BYTE1 rather than BYTE0.

For the case of a word read from an odd address, software must compensate for the following two effects:

- The low byte is in BYTE1 of the RFDATA register rather than in BYTE0.
- The high byte is not in the RFDATA register. Software must perform another read to read the high byte.



### *Saving and Restoring the State of the Read FIFO*

The following procedure can be used to save the state of the 32-bit Read FIFO:

1. Read the RFCON register and save it for the state restoration.
2. If the FIFO was in AUTO mode, wait until the FIFO is FULL by checking the RFCON register.
3. Read the RFCNTR register.
4. If the FIFO was in AUTO mode, subtract 8 from the value in RFCNTR. This is the FIFO Counter Value (i.e., the address of the next byte to be read from the SFB).

The FIFO can now be initialized and used for another transfer. Restoring the state of the FIFO is indistinguishable from beginning a new operation. It consists of beginning a new transfer using the 24-bit SFB address calculated in step 4.

## **5.7 Interrupts and Meta-Interrupts**

The 82750PD system of interrupts and meta-interrupts enables the 82750PD core and the host CPU to automatically learn of events concerning devices on the SFB. This system includes the SynchroLink registers (see Section 6.5), the interrupt registers in the core (see Section 3.14), and the interrupt registers in the host interface. This section describes the interrupt and meta-interrupt system, the associated UHBI registers, and a procedure for setting up interrupts.

### 5.7.1 The Interrupt/Meta-Interrupt System

SFBI events are communicated to the 82750PD by the SynchroLink. Table 5-27 lists the SFBI events that can be programmed to generate meta-interrupts to the 82750PD core and, possibly, an interrupt to the host. The second column describes how these events are manifested in the XCVR<sub>n</sub> and RSRV registers in the SynchroLink interface. Column three lists the register bits that can enable each interrupt event to generate a meta-interrupt.

Table 5-27. Enabling SFBI Events to Generate Meta-Interrupts

SFBI Event	Effect on SynchroLink Interface Registers	Enabling Bit	VBUS Code Group
The 82750PD has received and acknowledged an incoming message	Hardware sets VALID (bit 16) in RSRV register.	IINT (bit 24) in RSRV register	V1CODE
A message transmission by the 82750PD is complete	Hardware clears TXIPR (bit 16) in XCVR <sub>n</sub> register	TXINT (bit 24) in XCVR <sub>n</sub> register	V1CODE
A service request from the 82750PD was not accepted by the target device.	Hardware sets NACK (bit 17) in XCVR <sub>n</sub> register.	NINT (bit 25) in XCVR <sub>n</sub> register	V1CODE
A device that was sent a service request by the 82750PD has replied	Hardware sets COMPLETE (bit 19) in XCVR <sub>n</sub> register	CINT (bit 27) in XCVR <sub>n</sub> register	V1CODE
The 82750PD has received a message with a certain function code *	Hardware sets FMATCH (bit 18) in XCVR <sub>n</sub> register.	BINT (bit 26) in XCVR <sub>n</sub> register	V1CODE
The 82750PD has received a message with a certain function code *	Hardware sets FMATCH (bit 18) in XCVR <sub>n</sub> register	BINT2 (bit 29) in XCVR <sub>n</sub> register	V2CODE

\* The last two events are identical, but the meta-interrupts are enabled by different bits and generate different VBUS codes

Figure 5-12 shows a diagram of the interrupt and meta-interrupt system. The top portion shows the arrival paths of the SFBI events, which are identified by their enabling bits, as listed in Table 5-27. (See Sections 6.4.2 and 6.4.6 for descriptions of the XCVRn and RSRV registers, respectively.)

The SFBI events, represented by their enabling bits, are divided into two groups. V1CODE events consist of IINT, TXINT, NINT, BINT, and CINT. V2CODE events consist of BINT2 only. Whenever an enabled event in the V1CODE group occurs, a message is sent to the core. The message is the same, regardless of which event in the V1CODE group occurs. Similarly, if BINT2 is enabled and its corresponding event occurs, a message is sent to the core. The exact message is determined by the V1CODE and V2CODE fields of the MCFG register (Section 6.4.9). When setting up the interrupt system, the programmer can write into the V1CODE field any VBUS code listed in Table 5-28. The VBUS code is chosen according to the desired action(s) in response to a V1CODE event. A similar choice is made for V2CODE.

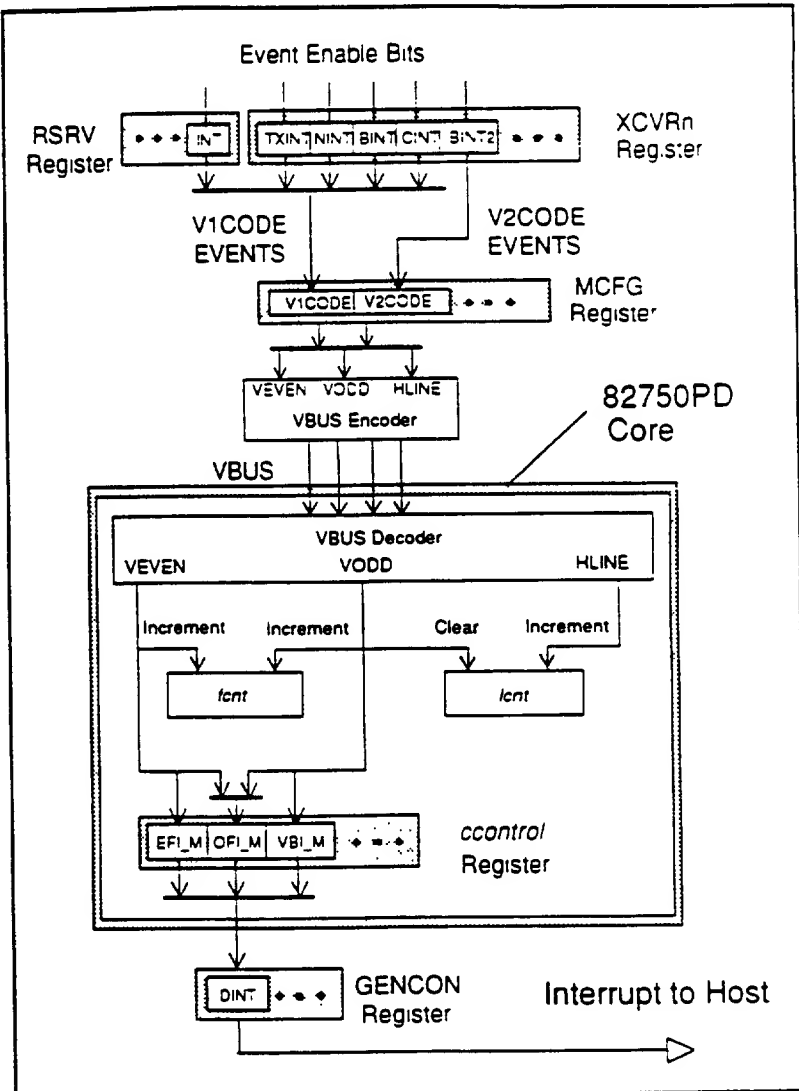


Figure 5-12. Diagram of the Meta-Interrupt System

Table 5-28. VBUS Codes

VBUS Code*	Name	Action
0000-1011	Reserved**	
1100	VODD	VBI interrupt, OF interrupt increment <i>fcnt</i> register, clear <i>lcnt</i> register
1101	VEVEN	VBI Int, EF Int, increment <i>fcnt</i>
1110	HLINE	increment <i>lcnt</i> register
1111	NULL	This is the idle state of the VBUS

\* The VBUS codes are written to the V1CODE field (bits 27-24) and the V2CODE field (bits 31-28) of the MCFG register.

\*\* Do not use Reserved codes

The selected code is sent over the VBUS to the core. The decoded signal (VEVEN, VODD, or HLINE) alters the *fcnt* and/or *lcnt* registers as shown in Figure 5-12. (These are the alterations described in Table 5-28.) This register alteration constitutes the meta-interrupt. It is software's responsibility to sense the meta-interrupt by polling the *fcnt* and *lcnt* registers.

The same VBUS code can also be programmed to trigger the 82750PD interrupt to the host. The bits in the *ccontrol* register enable the interrupt. A VEVEN code can trigger an even field interrupt (EFI\_E bit) and/or a vertical blanking interval interrupt (VBI\_E bit). A VODD code can trigger an odd field interrupt (bit OFI\_E) and/or a vertical blanking interval interrupt (VBI\_E bit). Only one of these enable bits should be set at a time. (Note that an HLINE code can trigger only a meta-interrupt.)

The 82750PD hardware sets fixed priorities for the generation of VBUS codes, as listed in Table 5-29.

Table 5-29. Priorities for Generation of VBUS Codes

Priority	VBUS code
Highest	Currently driven VBUS code
	V2CODE
Lowest	V1CODE

V1CODE requests are queued only 1 deep. If a second V1CODE request is generated before the first has been driven to the 82750PD core, then the second request is ignored. V2CODE requests are also queued only 1 deep.

The DINT bit in the GENCON (Section 5.7.1) register enables the single interrupt to the host. The interrupt output pin can be configured as three-state or open drain and with a desired polarity to meet the requirements of the particular bus type.

The DINT bit facilitates interrupt handling in both level-triggered and edge-triggered interrupt systems. When an interrupt occurs, the host's interrupt handler typically disables all interrupts while it processes the one that was just detected. If a second interrupt occurs while all of the interrupts are disabled, and if the interrupt system is level-triggered, it is detected when the interrupt handler re-enables the interrupts. However, in an edge-triggered interrupt system, such as the ISA bus, this second interrupt condition would be undetected because the interrupts are disabled when the edge occurs. This problem is solved with the use of the DINT bit (described in the following subsection). By employing the DINT bit, software can use the same approach for buses with level-triggered interrupts and edge-triggered interrupts. The DINT bit does this by giving software the ability to generate an interrupt edge, even when interrupt conditions occur very close to each other.

The next subsection describes the GENCON register, which includes the DINT bit, and the GENSTAT register, which indicates whether the 82750PD interrupt signal is asserted/deasserted. A second subsection describes how to set up the 82750PD interrupt.

## 5.7.2 General Control and General Status Registers

The General Control Register (REG6-BYTE2), shown in Figure 5-12, has only one functional bit, the Disable Interrupt bit (DINT).

Table 5-30. General Control Register

Mnemonic: GENCON			Address: REG6-BYTE2
Access: R/W			Reset State: 00h
Bit No.	7	6	5-0
Name	RSVD	DINT	RSVD

Bit	Name	Description
6	DINT	Disable Interrupt. Setting this bit disables the 82750PD interrupt. Clearing this bit enables the 82750PD to generate an interrupt.

Setting DINT disables the 82750PD interrupt. The DINT bit does not change the interrupt conditions; it only blocks the signal from being driven onto the interrupt line. (Note: DINT simply forces the interrupt inactive; DINT does not three-state the interrupt pin.) The interrupt handler should set DINT and then clear it after the system interrupt controller has been reset. If another interrupt was pending before the system interrupt controller was reset, that interrupt is reasserted.

The General Status Register, shown in Table 5-31, contains the Video Interrupt bit (VINT), a read-only bit that reflects the status of the interrupt signal

from the 82750PD. VINT is set whenever the interrupt signal from the 82750PD is asserted. To reset this bit, execute a read of the 82750PD Interrupt Flag register, which is located at '0xFE0100' (see Section 3.14).

Table 5-31. General Status Register

Mnemonic: GENSTAT

Address: REG6-BYTE3

Access: R/O

Reset State: 00h

Bit No.	7-2	1	0
Name	RSVD	VINT	RSVD

Bit	Name	Description
6	VINT	VINT. This bit is set whenever the 82750PD interrupt signal is asserted

### 5.7.3 Setting Up the 82750PD Interrupt

Setting up and handling the 82750PD interrupt involves the following registers:

- Three core registers:
  - Core Control Register (*ccontrol*, see Section 3.14.1).
  - Core Interrupt Flag Register (*cntrflag*, see Section 3.14.2).
  - Core Status Register (*cstatus*, see Section 3.14.3)
- General Control Register (GENCON, see Section 5.7.1).
- General Status Register (GENSTAT, see Section 5.7.1).
- Interrupt Configuration Register (INT CFG, see Section 5.8.4).



The following steps are a procedure for setting up the 82750PD interrupt:

1. Disable the 82750PD interrupt by setting the DINT bit (bit 6) in the GENCON register.
2. Configure the interrupt for the appropriate bus type by writing to the INT CFG register.
3. Enable or disable the individual interrupt sources by writing to the ccontrol register. The bits in ccontrol can be monitored by reading the cstatus register.
4. Enable the 82750PD interrupt by clearing the DINT bit (bit 6) in the GENCON register.

When an 82750PD interrupt occurs, the host CPU's interrupt handler can determine the source by reading the cirqflag register.

## 5.8 Configuration Registers

---

The 82750PD has configuration (CFG) registers that are accessed indirectly through two host I/O registers. This section contains bit descriptions of the CFG registers and describes how to access them.

The CFG registers are accessed indirectly through the CFG Register Number Register (CFGNUM) at REG6-BYTE0 and the CFG Data Register (CFGDATA) at REG6-BYTE1. Table 5-32 lists the CFG registers and their numbers.

Table 5-32. CFG Register Numbers and Names

CFG Register Number	CFG Register Name	CFG Register Number	CFG Register Name
00	General CFG	08	EMS CFG2
01	I/O Base CFG	09	RESERVED
02	Bus CFG	0A	RESERVED
03	INT CFG	0B	RESERVED
04	SFBI CFG0	0C	RESERVED
05	SFBI CFG1	0D	RESERVED
06	EMS CFG0	0E	RESERVED
07	EMS CFG1	0F	RESERVED

To access a CFG register, follow this sequence:

1. Write the encoded CFG register number to the CFGNUM register (REG6-BYTE0).
2. Read from or write to the CFGDATA register (REG6-BYTE1). The same CFG register can be repeatedly accessed via the CFGDATA register without writing again to the CFGNUM register.

The CFGNUM register must be setup before accessing the CFGDATA register. Software may not access the CFGNUM register and the CFGDATA register in the same instruction. Table 5-33 and 5-34 show the structure of the CFGNUM and CFGDATA registers.

Table 5-33. CFG Register Number Register

Mnemonic: CFGNUM

Address: REG6-BYTE0

Access: R/W

Reset State: 00h

Bit No.	7-4	3-0
Name	RSVD	R[3:0]

Bit No.	Name	Description
0	R0	Bit 0 (LSB) of the CFG register number
1	R1	Bit 1 of the CFG register number
2	R2	Bit 2 of the CFG register number
3	R3	Bit 3 (MSB) of the CFG register number
4-7	RSVD	

Table 5-34. CFG Register Data Register

Mnemonic: CFGDATA

Address: REG6-BYTE1

Access: R/W

Reset State\*

Bit No.	31-0
Name	—

\* This is not a physical register.

Bit No.	Name	Description
15-0		This register holds the data that is read from or written to a configuration register that is accessed via the CFGNUM register (REG6-BYTE0)

## 5.8.1 General CFG Register

Table 5-35. General Configuration Register

Mnemonic: CFGGEN

Address: CFG 00

Access: R/O

Reset State\*

Bit No.	7	6	5-3	2-0
Name	MTYP	RSVD	BTYP[5:3]	DID[2:0]

\* Determined by strapping options at reset.

Bit No.	Name	Description
2-0	DID[2:0]	Device ID. These bits define the device ID of the 82750PD. The bit values are determined from the configuration straps at reset.
5-3	BTYP[2:0]	Bus Type. These bits reflect the bus type. The bit values are determined from the configuration straps at reset. Strapping information is given in the data sheet.
6	RSVD	
7	MTYP	Memory Type. This bit reflects the memory type. The value is determined from the configuration straps at reset. Strapping information is given in the data sheet.

## 5.8.2 I/O Base CFG

Table 5-36. I/O Base CFG Register

Mnemonic: CFGBASE

Address: CFG 01

Access: R/O

Reset State: See bit definitions

Bit No.	7-0
Name	IOB[7:0]

Bit No.	Name	Description
7-0	IOB[7:0]	These bits indicate the state of the IO_BASE 'POS' field. See Section 5.2, "Host Interface Address Configuration" for how these bits are reset and/or changed.

### 5.8.3 Bus CFG Register

The 82750PD has a Bus CFG Register (CFGBUS) with the structure shown in Table 5-37. Following this table are tables describing the CFGBUS register contents for each bus type supported by the 82750PD.

Table 5-37. Bus CFG Register

Mnemonic: CFGBUS		Address: CFG 02
Access: R/W		Reset State: 00h
Bit No.	7-0	
Name	BCFG[7:0]	

Bit No.	Name	Description
7-0	BCFG[7:0]	These configuration bits depend on the bus type

**ISA BCFG[7:0].** Table 5-38 describes the bus configuration bits for the ISA bus. The default ISA configuration runs standard 8-bit memory cycles and standard 8-bit I/O cycles. Zero wait state and 16-bit cycles are turned off.

**Table 5-38. Configuration Bits for the ISA Bus.**

Bit No.	Name	Description
0	BCFG[0]	Setting BCFG[0] disables the addition of wait states when the host writes fields that affect the address decoder.
1	BCFG[1]	Setting BCFG[1] enables the issuance of MCS16. This effectively enables 16-bit memory cycles. (The default is 8-bit memory cycles.) Bits 1 and 4 must have the same value.
2	BCFG[2]	Setting BCFG[2] enables the insertion of one additional MCLK period in the time necessary for a command to be recognized as active.
3	BCFG[3]	Setting BCFG[3] enables the issuance of NOWS. This effectively enables compressed cycles.
4	BCFG[4]	Setting BCFG[4] enables the issuance of IO16. This effectively enables 16-bit I/O cycles. (The default is 8-bit I/O cycles.) Bits 1 and 4 must have the same value.
5	BCFG[5]	BCFG[5] must always be clear. (Setting this bit may hang the system.)
6	BCFG[6]	BCFG[6] must always be clear. (Setting this bit may hang the system.)
7	BCFG[7]	BCFG[7] must always be clear. (Setting this bit may hang the system.)

**EISA BCFG[7:0].** Table 5-39 describes the bus configuration bits for the EISA bus. The default EISA configuration runs single standard 32-bit memory cycles and single standard 8-bit I/O cycles with rescinding output buffers. Compressed, burst, and 16-bit I/O cycles are turned off.

Table 5-39. Configuration Bits for the EISA Bus.

Bit No.	Name	Description
0	BCFG[0]	Setting BCFG[0] disables the addition of wait states when the host writes fields that affect the address decoder.
1	BCFG[1]	Setting BCFG[1] enables issuance of SLBURST for write cycles. This effectively enables write burst cycles.
2	BCFG[2]	Setting BCFG[2] enables issuance of SLBURST for read cycles. This effectively enables read burst cycles.
3	BCFG[3]	Setting BCFG[3] enables the issuance of NOWS. This effectively enables compressed cycles.
4	BCFG[4]	Setting BCFG[4] enables issuance of IO16. This effectively enables 16-bit I/O cycles. (The default is 8-bit I/O cycles.)
5	BCFG[5]	Setting BCFG[5] disables the use of rescinding output buffers.
6	BCFG[6]	BCFG[6] must always be clear. Setting this bit may hang the system.
7	BCFG[7]	BCFG[7] must always be clear. Setting this bit may hang the system.

**PCI BCFG[7:0].** Table 5-40 describes the bus configuration bits for the PCI bus. The default PCI configuration runs single memory cycles and I/O cycles. (STOP is generated after the first data cycle.) If the master generates a non-burst cycle, a read executes in a minimum of 4 clock periods, and a write executes in a minimum of 3 clock periods. Burst and 2-clock write cycles are turned off.

**Table 5-40. Configuration Bits for the PCI Bus**

Bit No.	Name	Description
0	BCFG[0]	Setting BCFG[0] disables the addition of wait states when the host writes fields that affect the address decoder
1	BCFG[1]	Setting BCFG[1] enables burst write cycles. If this bit is clear, STOP is issued after the first data cycle
2	BCFG[2]	Reserved
3	BCFG[3]	Reserved
4	BCFG[4]	Setting BCFG[4] enables toggle-mode read burst cycles. If this bit is clear, STOP is issued after the first data cycles
5	BCFG[5]	Setting BCFG[5] enables incrementing read burst cycles. If this bit is clear, STOP is issued after the first data cycles
6	BCFG[6]	Setting BCFG[6] enables decrementing read burst cycles. If this bit is clear, STOP is issued after the first data cycles
7	BCFG[7]	Setting BCFG[7] allows address decoding and the generation of TRDY in the same clock period that frame is asserted. This effectively enables two clock write cycles



## Universal Host Bus Interface

**VL-BCFG [7:0].** Table 5-41 describes the bus configuration bits for the VL-Bus. The default VL-Bus configuration runs single memory cycles and I/O cycles (BRDY is not generated). If the master generates a non-burst cycle, a read executes in a minimum of 4 clocks, and a write executes in a minimum of 3 clocks. Burst and 2-clock write cycles are turned off.

Table 5-41. Configuration Bits for the VL-Bus

Bit No.	Name	Description
0	BCFG[0]	Setting BCFG[0] disables the addition of <del>wait</del> states when the host writes fields that affect the address decoder.
1	BCFG[1]	Setting BCFG[1] enables burst write cycles. If this bit is clear BRDY is not generated and write cycles respond with LRDY
2	BCFG[2]	Reserved
3	BCFG[3]	Reserved
4	BCFG[4]	Setting BCFG[4] enables toggle mode read burst cycles. If this bit is clear, BRDY is not generated and read cycles respond with LRDY
5	BCFG[5]	Setting BCFG[5] enables the removal of the forced address turnaround cycles on the multiplexed A/D bus
6	BCFG[6]	Setting BCFG[6] enables the removal of the forced data turnaround cycles on the multiplexed A/D bus
7	BCFG[7]	Setting BCFG[7] enables the removal of the extra cycle inserted to allow additional address decode time

### 5.8.4 Interrupt CFG Register

The Interrupt Configuration Register (Table 5-43) has bits for configuring the 82750PD interrupt. Table 5-42 lists the values of the VIP bit, which determines the interrupt signal polarity, and the VIT bit, which selects a totem pole or open drain interrupt output.

Table 5-43. INT CFG Register

Mnemonic: CFGINT

Address: CFG 03

Access: See bit descriptions.

Reset State: See bit descriptions.

Bit No.	7	6	5	4	3	2	1	0
Name	MEMEN	IOEN	IOB8	HALT	RSVD	VIP	VIE	VIT

Bit No.	Name	Description
0	MEMEN	Memory Enable. This R/O bit reflects the state of the MEM_ENABLE 'POS' field (See Section 5.2 for a description of this bit.)
1	IOEN	IO Enable. This R/O bit reflects the state of the IO_ENABLE 'PCS' field (See Section 5.2 for a description of this bit.)
2	IOB8	I/O Base. This R/O bit indicates the state of the 9th bit of the IO_BASE 'POS' field. This bit is latched from a strap during reset.
3	HALT	HALT. This R/W bit is used to control the 82750PB core. Setting HALT causes the PB core execution to halt. Reset state = 0
4	RSVD	
5	VIP	VIP. This R/W bit is the interrupt level polarity bit for 82750PB interrupts. If VIP is set, the interrupt is active high. If VIP is clear, the interrupt is active low. Reset state = 0
6	VIE	VIE. This R/W bit is the Interrupt level enable bit for 82750PB interrupts. If VIE is set, the PB interrupt is presented on the IRQ pin. If VIE is clear, the PB interrupt is three-stated. Reset state = 0
7	VIT	VIT. This R/W bit is the Interrupt level type bit for 82750PB interrupts. If VIT is set, the PB interrupt is totem pole output. If VIT is clear, the PB interrupt is open drain output. Reset state = 0

Table 5-44. Values of Interrupt Configuration Bits

Bus Type	MEMEN	IOEN	VIE	VIT
ISA	1	1	1	1
EISA	= ENABLE bit in POS2 register	= ENABLE bit in POS2 register	1	1
Micro Channel	= CDEN bit (bit0) in POS2 register	= CDEN bit (bit0) in POS2 register	0	0
PCI	= MEMEN bit (bit 1) in POS2 register	= IOEN bit (bit 0) in POS2 register	0	0
VL-Bus	1	1	1	1

## 5.8.5 SFBI CFG Registers

Tables 5-45 and 5-46 show the Shared Frame Buffer Interconnect CFG registers. CFGSFBI4 and CFGSFBI5.

Table 5-45. SFBI CFG 04 Register

Mnemonic: CFGSFBI4

Address: CFG 04

Access: R/W

Reset State: 00h

Bit No.	7	6	5	4	3	2	1	0
Name	PAGE	REFEN	HCAS	HLCH	RSVD	MSIZ	MSZ	MSZ0

Bit No.	Name	Description
2-0	MSIZ[2:0]	<b>Memory Size.</b> This 3-bit R/W field defines the size of the Shared Frame Buffer (SFB). <b>MSIZ[2:0] SFB Size</b> 0 1 0    1MB (also configures SFBC as 32-bit) 0 1 1    2MB 1 0 0    4MB 1 1 0    6MB 1 1 1    8MB <i>Note: All other values are illegal. (Illegal values produce memory cycle and "readvs" but the data is not guaranteed)</i>
3	RSVD	
4	HLCH	<b>Half Latch.</b> If this R/W bit is set, the data latch is closed one-half clock early
5	HCAS	<b>Half CAS.</b> If this R/W bit is set, the CAS signal is activated one-half clock early
6	REFEN	<b>Refresh Enable.</b> Setting this R/W bit enables the SFB interface to cause a CAS-before-RAS refresh cycle in response to 82750PD refresh cycles
7	PAGE	<b>Page.</b> This bit determines whether the 82750PD leaves the SFB page open or closes it while the 82750PD owns the SFB. Setting this bit leaves the SFB page open (RAS low). Clearing this bit closes the SFB page (RAS high)

Table 5-46. SFBI CFG 05 Register

Mnemonic: CFGSFBI5

Address: CFG 05

Access: R/W

Reset State: 00h

Bit No.	7	6	5	4	3	2	1	0
Name	MMPR1	MMPR0	QUASI	CCNT4	CCNT3	CCNT2	CCNT1	CCNT0

Bit No.	Name	Description
4-0	CCNT[4:0]	These 5-bit R/W field that defines the maximum number of SFBC clocks (0 - 32) that the 82750PD is allowed to issue before giving up the SFBC in response to GRANT being deasserted. If GRANT is deasserted, the 82750PD continues using the SFBC (if it has any outstanding requests) until it has issued the maximum number of clocks. At the end of this time interval, the 82750PD completes any current bus cycle and then release the SFBC.
5	Quasi	This R/W bit determines whether the GRANT signal from the SFBC arbiter is synchronous or asynchronous. 0 = Synchronous 1 = Asynchronous
7-6	MMPR[1:0]	This 2-bit R/W field that defines the priority the 82750PD uses when requesting the SFBC. <u>MMPR[1:0] Priority</u> 0 0 High Priority 0 1 Medium Priority 1 0 Low Priority 1 1 No Priority (ILLEGAL! If No Priority is programmed and any memory cycle is attempted, the 82750PD HANGS.)

5.8.6 "EMS" CFG Registers

The "EMS" CFG registers are in Section 5.4.2.

5.8.7 Reserved CFG Registers

CFG registers 09h-0Fh are Reserved. These registers appear in the CFG register map (see Table 5-32), but no physical registers exist. Data written to these reserved CFG registers is ignored. Data read from them is undefined.

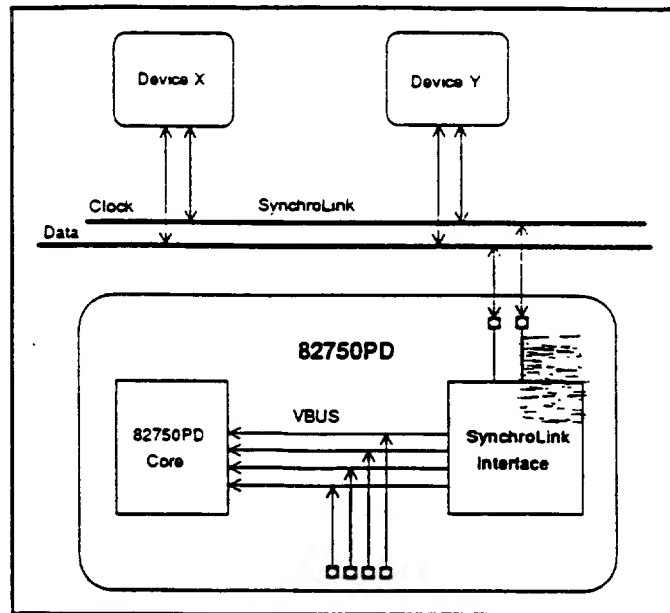
### 6.1 Introduction

---

The multimedia system can function successfully only if the devices on the Shared Frame Buffer Interconnect (SFBI) are synchronized. For example, when the 82750PD has decompressed a frame for display, it should notify the graphics processor that the frame is ready. In reply, the graphics processor should signal the 82750PD when it has finished writing the frame. SynchroLink is a serial two-wire bus, which, together with the Synchrolink interface, supports these communications.

Events that should be communicated to devices on the SFBI are *SFBI events*. A message regarding an SFBI event is sent in the form of a data packet on the SynchroLink. Figure 6-1 shows the connection of the 82750PD and two other devices to the SynchroLink. The bus activities are directed by one of the devices, the arbiter, which invites each device in turn to transmit its message.

The 82750PD writes to registers in the SynchroLink interface to prepare a data packet that is sent over the SynchroLink. In return, data packets from other devices update the SynchroLink interface registers. Some incoming data packets generate meta-interrupts, which are sent to the 82750PD core via the four-wire, unidirectional VBUS. The VBUS is also connected directly to package pins for test purposes.



**Figure 6-1. The SynchroLink with Devices Connected**

This chapter describes communications via the SynchroLink. Sections 6.2 through 6.4 describe the protocol for the bus communications and the registers in the SynchroLink interface. Section 6.5 describes meta-interrupts and the VBUS.

**NOTE:**

*Software developers should initialize all registers prior to use.*

## 6.2 Arbiter, Source, and Target

Devices on the SFBI assume different roles when communicating over the SynchroLink. One device is the *arbiter*, which provides a stable clock and directs the communications. The arbiter sends a series of SynchroLink messages that invite each device, in turn, to use a brief time slot to send its message. Any device on the bus should be capable of functioning as the arbiter. During device initialization, software designates one device to be the arbiter.

All devices on the bus monitor the invitations from the arbiter. When a device receives its invitation, it can decline or it can choose to accept the invitation and thereby become the *source*. The source is given temporary control of the bus to send a message. For example, a display processor can broadcast to all devices that it has just finished "painting" the display screen from top to bottom.

In a different type of message, the source can request a service from a particular device, the *target*. The target acknowledges that the request was received. In a subsequent message, the target becomes the source and sends a the requesting device a reply saying that the service was performed successfully or unsuccessfully. For example, the 82750PD could send a "Go Blt" request to the graphics controller and then receive a "Blt Done." The reply could be programmed to generate a meta-interrupt to the 82750PD core. (See Section 6.5 for a description of meta-interrupts.)

The 82750PD SynchroLink interface is capable of sending messages to itself. If an outgoing message is detected as being a message that the 82750PD should receive, then it is received properly. This loopback capability applies to all types of messages and requires no special bit settings.



### 6.3 SynchroLink Data Packets

This section describes SynchroLink *data packets*, with attention to the roles of the arbiter, the source, and the target. Following subsections treat the data packets in more detail and describe how to program the SynchroLink registers to generate the desired data packets.

#### 6.3.1 Arbiter, Source, and Target Interactions

A typical data packet, shown in Figure 6-2, begins when the arbiter transmits an *invitation*, which is composed of a start bit (bit 0) followed by a 3-bit identification number, the *message ID*, for the invited device. (Each device on the SynchroLink has a unique message ID). The arbiter performs a switch over on cycle 4, leaving the bus in the de-asserted state. If the device with the matching message ID wants to send a message, it asserts an Invitation Acknowledge (bit 5) to indicate that it is now the source and will commence transmission of a *signal event* (bits 6-13, 15-22). Bit 23 is a switch over bit that allows the arbiter to take over the bus and begin the next cycle of invitations.

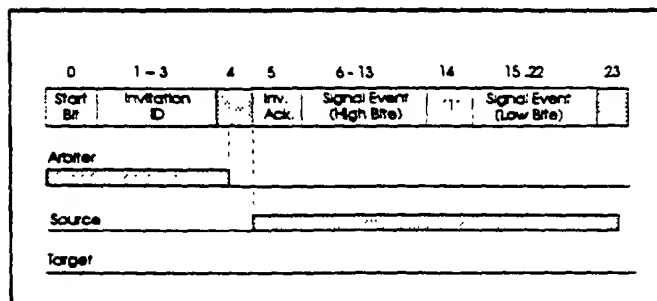


Figure 6-2. Data Packet - Source Accepts

In some cases, a signal event from the invited source requires an acknowledgment from the target (destination device) of the signal event (see Figure 6-3). In this case, the Service Acknowledge bit is driven by the target at bit 22 of the same packet. Bit 21 is then used as a switch over time for the source of the signal event to release the bus to the target. A service request requires an acknowledgment because the target may have very limited queuing capabilities and may miss a signal event if it is too busy to accept the service request. If a request is not acknowledged, the requester can retry each time it is invited to use the bus until the request is acknowledged.

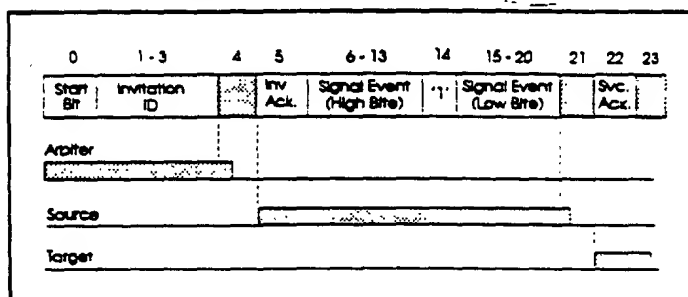


Figure 6-3. Data Packet - Source Accepts and Target Acknowledges

A SynchroLink transaction requires one or two data packets (messages). If the source merely wants to inform all of the other devices that it has performed some operation, one packet is sufficient. If Device X asks Device Y to perform a service, Device Y acknowledges the request in the first packet. In a second packet Device Y informs Device X that the service is completed successfully or unsuccessfully.

Most of the time, the SynchroLink carries only circulating invitations from the arbiter with no device actually accepting the invitations (see Figure 6-4). In these cases, the signal event portion of the packet is skipped. It is the responsibility of each device on the bus to monitor the Invitation Acknowledge bit of each invitation to determine when to begin looking for the next start bit.

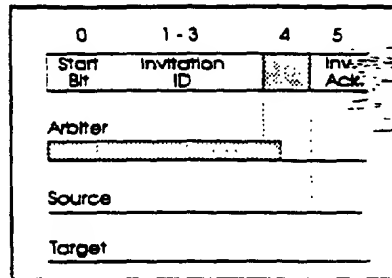


Figure 6-4. Data Stream - Invitation Not Acknowledged

## 6.3.2 Data Packet Fields

This section defines the data packet fields, which are illustrated in Figure 6-4 and described in subsequent paragraphs.

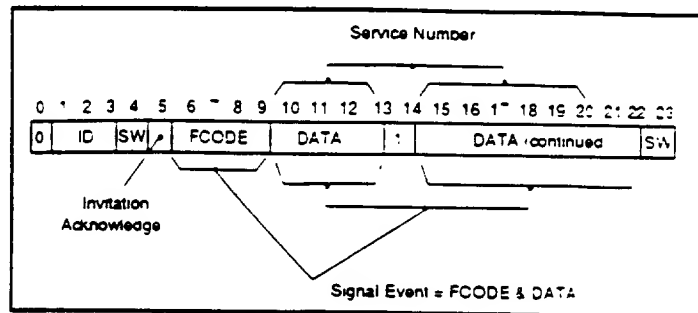


Figure 6-5. Data Packet Format

**Start Bit** (bit 0).

**Message ID** (bits 1-3). This field, written by the arbiter, contains the ID of the device that is invited to take over the SynchroLink. If that device accepts the invitation, it becomes the source.

**SW** (Switch over, bit 4). This is a switch over bit. The bus is de-asserted while control is transferred from the arbiter to the source.

**Invitation Acknowledge** (bit 5). The invited device sets this bit to acknowledge the invitation and to announce that it has become the source.

**FCODE** (bits 6-9). The *function code*, which is written by the source, specifies an action or a type of information. Of the sixteen possible function codes, only two are predefined: *service request* and *service completion*. These are described in Section 6.3.3. Software can define other function codes.

**Service Number** (bits 10-13, 15-20). When the source asks a target device to perform a service, it writes a *service number* to this 10-bit field. The service number specifies two things: the target and the service to be performed by that target.

For the 82750PD, the Match Service Register (MSRV, see Section 6.4.8) specifies a range of values within the 1024 possible values in the service number field. The 82750PD monitors the service number of any service request packet on the bus. If the service number is within the range specified by the MSRV register, the 82750PD is the target and must respond. From the specific value it determines which service is requested to perform. Each of the devices on the bus recognizes its own unique range of service numbers.

**DATA** (bits 10-13, 15-22 or 15-20). The *data* field, which is written by the source, includes the bits of the service number plus bits 21 and 22. However, if the source requires an acknowledgment from the target (as in the example of Figure 6-3), the data field is shortened to accommodate a switch over bit (bit 21) and an acknowledge bit (bit 22) from the target.

**Signal Event** (bits 6-13, 15-22, or 15-20). The *signal event* is composed of the function code and the data or the service number. Examples of signal events are in Table 6-1.

**SW** (Switch over, bit 23). This is a switch over bit. The bus is de-asserted while control is transferred from the source to the arbiter or from the target to the arbiter.

Table 6-1. Examples of Signal Events

Function Code (bits 6-9)	Data (bits 10-13, 15-20)*	Bit 21	Bit 22
Service Request = 0Eh	A ten-bit Service Number (bits 10-13, 15-20)	Switch over	Acknowledge receipt of message
Service Completion = 0Fh (always paired with service request)	10-bit Service Number	Unused	Service successful
Audio Record Sync *	12-bit Time Stamp		
Audio Playback Sync *	12-bit Time Stamp		
Graphics Scan Line Count *	12-bit Line Number		
Video Scan Line Count *	12-bit Line Number		

\* These examples have no predefined function codes

### 6.3.3 Service Requests

---

When a source device wants another device to perform a service, the source sends a *service request* data packet, which consists of the following elements:

- a function code: FCODE = 0Eh
- a 10-bit service number (bits 10-13, 15-20).
- a switch over bit (bit 21) to release the bus to the target. (The source de-asserts the bus and then releases it to a high impedance state.)

The target sets bit 22 in the ~~same~~ data packet to acknowledge receipt of the service request. Table 6-2 gives a detailed description of a service request data packet.

Table 6-2. Service Request Data Packet Definition

Bit(s)	Description
0	Start bit <ul style="list-style-type: none"> <li>Always set</li> <li>Driven by arbiter</li> <li>All devices synchronize to the leading edge of this bit</li> </ul>
1-3	Message ID <ul style="list-style-type: none"> <li>Driven by arbiter</li> <li>This is the message ID of the device that has control during bits 5-22.</li> <li>Bit 1 is the MSB of the ID</li> </ul>
4	Switch Over <ul style="list-style-type: none"> <li>First driven inactive by arbiter then released to high impedance</li> </ul>
5	Invitation Acknowledge <ul style="list-style-type: none"> <li>Set by the device with matching ID to indicate that it will send a signal event.</li> <li>If not driven, this bit defaults to '0' and bits 6-23 are skipped</li> </ul>
6-9	Function Code <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>Bit 6 is the MSB.</li> <li>= 0Eh</li> </ul>
10-13	Service Number (most significant 4-bits) <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>Contains the high 4-bits of the 10-bit data field</li> <li>Bit 10 is the MSB</li> </ul>
14	Event in Progress <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>Always set</li> <li>Used only to prevent a "break" (ten consecutive '0' bits) from being buried in the signal event</li> </ul>
15-20	Service Number (least significant 6 bits) <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>These are the low 6 bits of the 10-bit data field</li> <li>Bit 20 is the LSB</li> </ul>
21	Switch Over <ul style="list-style-type: none"> <li>Cleared by the device with the matching ID then released to high impedance</li> </ul>
22	Service Ack <ul style="list-style-type: none"> <li>Set by the target to acknowledge receipt of the request</li> </ul>
23	Switch Over <ul style="list-style-type: none"> <li>Cleared by the target then released to high impedance</li> </ul>



#### 6.3.4 Service Completion Message

---

A *service completion* message is a reply from the device that received the service request to the device that sent the request. A service completion data packet consists of these elements:

- A function code = 0Fh
- The same 10-bit service number that was in the service request. This indicates that the target is responding to that service request.
- An unused bit (bit 21).
- A Service Successful flag (bit 22) indicating that the service was performed successfully (= 1) or unsuccessfully (= 0).

Table 6-3 gives a detailed description of a service completion data packet.

Table 6-3. Service Completion Data Packet Definition

Bit(s)	Description
0	Start bit <ul style="list-style-type: none"> <li>Always set</li> <li>All devices synchronize to the leading edge of this bit</li> <li>Driven by the arbiter</li> </ul>
1-3	Message ID <ul style="list-style-type: none"> <li>Indicates which device has control during bits 5-22</li> <li>Bit 1 is the MSB of the ID</li> <li>0-7 (bits 1-3 = 000-111) are available as valid message ID</li> <li>Driven by arbiter</li> </ul>
4	Switch Over <ul style="list-style-type: none"> <li>First driven inactive by arbiter then released to high impedance</li> </ul>
5	Invitation Acknowledge <ul style="list-style-type: none"> <li>Set by the device with matching ID to indicate that it will send a signal event</li> <li>If not set, this bit defaults to '0' and bits 6-23 are skipped</li> </ul>
6-9	Function Code (high byte) <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>Bit 6 is the MSB</li> <li>= 0Fh</li> </ul>
10-13	Service Number (most significant 4-bits) <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>These are the high 4-bits of the 10-bit data field</li> <li>Bit 10 is the MSB</li> </ul>
14	Event in Progress <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>Always set</li> <li>Used only to prevent a "break" (ten consecutive '0' bits) from being buried in the signal event</li> </ul>
15-20	Service Number (least significant 6 bits) <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>These are the low 6-bits of the 10-bit data field</li> <li>Bit 20 is the LSB</li> </ul>
21	Unused <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>The value of this bit is a "don't care"</li> </ul>
22	Service Successful <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>Set by the target of the request if the service was performed successfully</li> </ul>
23	Switch Over <ul style="list-style-type: none"> <li>Cleared by the device with the matching ID then released to high impedance</li> </ul>

### 6.3.5 Broadcast Messages

Table 6-4 describes the data packet for a broadcast message. The source can select a function code value in the range 00h-0Dh (i.e., any value other than those for a service request (0Eh) or a service completion (0Fh)). A broadcast message can be received by any device(s) that are set up to receive the same function code value (see Section 6.4.3).

Table 6-4. Other Function Code Data Packet Definition

Bit(s)	Description
0	Start bit: <ul style="list-style-type: none"> <li>Always set</li> <li>All devices synchronize to the leading edge of this bit</li> <li>Driven by arbiter</li> </ul>
1-3	Message ID <ul style="list-style-type: none"> <li>Indicates which device has control during bits 5-22</li> <li>Bit 1 is the MSB of the ID</li> <li>0-7 (bits 1-3 = 000 - 111) are available as valid message IDs</li> <li>Driven by the arbiter</li> </ul>
4	Switch Over <ul style="list-style-type: none"> <li>Cleared by the arbiter and then released to high impedance</li> </ul>
5	Invitation Acknowledge <ul style="list-style-type: none"> <li>Set by the device with the matching ID to indicate that it will send a signal event</li> <li>If not set, this bit defaults to '0' and bits 6-23 are skipped</li> </ul>
6-9	Function Code <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>Bit 6 is the MSB</li> <li>The value is in the range 00h-0Dh</li> </ul>
10-13	Signal Event (most significant 4-bits) <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>Contains the most significant 4-bits of the data field</li> <li>Bit 10 is the MSB</li> </ul>
14	Event in Progress <ul style="list-style-type: none"> <li>Always set</li> <li>Used only to prevent a "break" (ten consecutive '0' bits) from being buried in the signal event</li> </ul>
15-22	Signal Event (low byte) <ul style="list-style-type: none"> <li>Only transmitted if the invitation acknowledge was asserted</li> <li>This is least significant byte of the data field</li> <li>Bit 22 is the LSB</li> </ul>
23	Switch Over <ul style="list-style-type: none"> <li>Cleared by the device with the matching ID and then released to high impedance</li> <li>A start bit can begin in the next clock cycle to initiate another invitation</li> </ul>

## 6.4 SynchroLink Registers and Operations

This section describes the SynchroLink operations and the SynchroLink registers. The first subsection introduces the registers. Following subsections give more detailed descriptions of the operations and the associated registers.

### 6.4.1 SynchroLink Registers

The SynchroLink interface includes sixteen registers that control the SynchroLink operations. These registers, listed in Table 6-5, are mapped into the 82750PD memory address space (see Table 2-1), where they can be accessed by the 82750PD core and the host.

Table 6-5. SynchroLink Register Map

Register Name	Mnemonic	Memory Address
Transceiver Register	XCVR0	00FF0000-00FF0003
"	XCVR1	00FF0004-00FF0007
"	XCVR2	00FF0008-00FF000E
"	XCVR3	00FF000C-00FF000F
"	XCVR4	00FF0010-00FF0013
"	XCVR5	00FF0014-00FF0017
"	XCVR6	00FF0018-00FF001B
"	XCVR7	00FF001C-00FF001F
"	XCVR8	00FF0020-00FF0023
"	XCVR9	00FF0024-00FF0027
"	XCVRA	00FF0028-00FF002B
"	XCVRB	00FF002C-00FF002F
Message Status Register	MSTATUS	00FF0030-00FF0033
Receive Service Register	RSRV	00FF0034-00FF0037
Match Service Register	MSRV	00FF0038-00FF003B
Message Configuration Register	MCFG	00FF003C-00FF003F
Reserved	—	00FF0040-00FFFFFF

The Transceiver Registers (XCVR<sub>n</sub>,  $n = 00h-0Bh$ ) perform and control operations, provide status information, and hold received data or data to be sent over the SynchroLink.

The Message Status Register (MSTATUS) provides information on the status of all of the XCVR<sub>n</sub> registers. The Match Service Register (MSRV) defines the range of service numbers to which the 82750PD should respond. The Message Configuration Register (MCFG) is used to configure the SynchroLink interface and to specify the use of the internal VBUS.

## 6.4.2 Transceiver Registers (XCVRn)

All of the Transceiver Registers have the format shown in Table 6-6. The paragraphs following describe the register bits. Comprehensive bit descriptions are in Appendix A.

**NOTE:**

*Software developers should initialize all registers prior to use.*

Table 6-6. Transceiver Registers

Mnemonic: XCVRn, n = 00h-0Bh

Address: 00FF0000h-00FF002Fh

Access: R/W

Reset State: See bit descriptions.

Bit No.	31	30	29	28	27	26	25	24
Name	MSG	RXEN	BINT2	RSVD	CINT	BINT	NINT	TXIPR

Bit No.	23-22	21	20	19	18	17	16
Name	RSVD	EXTRA	SUCCESS	COMPLETE	FMATCH	NACK	TXIPR

Bit No.	15-12	11-0
Name	FCODE	DATA

**DATA** (bits 11-0; uninitialized at reset). These bits correspond to the data bits in the data packet (packet bits 10-13, 15-22). When the register is in the broadcast transmit mode (RXEN = 0), software can write these bits. When the register is in the broadcast receive mode (RXEN = 1), hardware can write these bits.

**FCODE** (function code, bits 15-12; uninitialized at reset). This field contains the function code.

**TXIPR** (Transmit Initiate, bit 16; reset state = 0). Software sets this bit to initiate a transmission, i.e. to write the FCODE and DATA fields onto the

SynchroLink. Hardware clears this bit when the transmission is complete, regardless of whether the transmission is acknowledged.

**NACK** (Not Acknowledged, bit 17; reset state = 0) This bit is used only in a service-request/service-completion operation (SMSG = 1). Hardware sets this bit in the source's register to indicate that the target does not acknowledge the service request.

**FMATCH** (Function Code Match, bit 18; reset state = 0). This bit is used by a device that is in broadcast receive mode (RXEN = 1). Software clears this bit when it begins listening for a certain type of broadcast message (i.e., for a certain function code). Hardware sets this bit to indicate that a message with the specified function code has been received. To listen for more than one function code, the device must set up a separate register for each code.

**COMPLETE** (bit 19; reset state = 0). This bit is used only in a service-request/service-completion transaction (SMSG = 1). Software must clear this bit in the source's register when it transmits a service request (i.e., sets TXIPR). Hardware sets the COMPLETE bit when a corresponding completion message is received, i.e. when:

- The FCODE in the service completion message is 0Fh. and
- The service number (packet bits 10-13, 15-20) in the completion message matches the service number in the transceiver register that was used to send the service request.

**SUCCESS** (bit 20; reset state = 0). This bit is set by hardware to indicate that a service has been completed successfully. (When a service completion message causes the COMPLETE bit to be set, the value of the SUCCESS bit is latched from packet bit 22 of the message.)



**EXTRA** (bit 21; reset state = 0). This bit is currently undefined. In a read operation its state is unknown. It should be written as '0'.

**TXINT** (bit 24), **NINT** (bit 25), **BINT** (bit 26), **CINT** (bit 27), **BINT2** (bit 29): reset state = 0 for all of these bits. Each of these bits enables the SynchroLink interface to generate a meta-interrupt in response to an *SFBI event*, i.e., an event involving a device on the SFBI. Such an event is manifested by a bit being set or cleared in an  $\text{XCVR}_n$  register. This bit activity causes the SynchroLink interface to send a VBUS code to the 82750PD core, provided that the corresponding enabling-bit is set. Table 6-7 lists the bit activity in  $\text{XCVR}_n$  that signals the occurrence of an SFBI event, the corresponding enabling bits in  $\text{XCVR}_n$ , and the VBUS codes generated. Section 6.5 has further information on meta-interrupts and VBUS codes. Section 5.7 describes the system of interrupts and meta-interrupts.

Table 6-7. Enabling SFBI Events to Generate VBUS Codes

Bit Activity in $\text{XCVR}_n$ Register	Enabling Bit in $\text{XCVR}_n$	VBUS Code
Hardware clears TXIPR bit (bit 16)	TXINT (bit 24)	V1CODE
Hardware sets NACK bit (bit 17)	NINT (bit 25)	V1CODE
Hardware clears FMATCH bit (bit 18)	BINT (bit 26)	V1CODE
Hardware clears COMPLETE bit (bit 19)	CINT (bit 27)	V1CODE
Hardware sets FMATCH bit (bit 18)	BINT2 (bit 29)	V2CODE

**RXEN** (bit 30) and **SMSG** (bit 31). This pair of bits controls the mode of this register. The RXEN bit determines whether the DATA field of this register can be written by the 82750PD software or by the SynchroLink hardware. The SMSG bit determines whether the register is in the broadcast mode (send/receive) or the service mode (service request/service completion). Table 6-8 lists the

register control bits that are used for each of the transmit/receive modes. Note that while these bits have the same values for service request and service completion, the two modes can be distinguished by the FCODE fields: 0Eh = service request, 0Fh = service completion.

Table 6-8. XCVRn Register Modes

Register Mode	SMSG	RXEN	Register Control Bits
Broadcast Transmit	0	0	TXIPR, TXINT
Broadcast Receive	0	1	FMATCH, BINT, BINT2
Send Service Request	1	0	TXIPR, TXINT, NACK, NINT
Receive Service Completion	1	0	CINT, SUCCESS, COMPLETE, EXTRA
Reserved	1	1	

Do not change the SMSG bit or the RXEN bit unless MSG\_EN = 0 and DMSG\_EN = 0 in the Message Configuration Register (see Section 6.4.9). Changing the SMSG or RXEN bit when MSG\_EN = 1 or DMSG\_EN = 1 may corrupt a message on the SynchroLink.

Table 6-9 shows the XCVRn registers in each of the four transmit/receive modes. The mode is determined by the SMSG and RXEN bits (along with the FCODE for service request and service completion). For any mode, the bits marked with an asterisk are operational: they function according to their bit descriptions. For a register read, the values of the shaded bits are meaningless. When the register is written, the shaded bits should be cleared.

Table 6-9. XCVRn Settings for the Four Message Modes

**Broadcast Transmit**

MSG	RXEN	BINT2	CINT	BINT	NINT	TXINT	SUCCSS	CMPLT	FMATCH	NACK	TXIPR
0	0					*					*

**Broadcast Receive**

MSG	RXEN	BINT2	CINT	BINT	NINT	TXINT	SUCCSS	CMPLT	FMATCH	NACK	TXIPR
0	1	*		*					*		

**Send Service**

MSG	RXEN	BINT2	CINT	BINT	NINT	TXINT	SUCCSS	CMPLT	FMATCH	NACK	TXIPR
1	0				*	*				*	*

**Receive Completion**

MSG	RXEN	BINT2	CINT	BINT	NINT	TXINT	SUCCSS	CMPLT	FMATCH	NACK	TXIPR
1	0		*				*	*			

- \* For a register read, the value of this bit is meaningless  
For a register write, this bit must be cleared.
- \* This bit is operational for this mode

The XCVRn registers can generate multiple simultaneous requests for transmission. The arbitration of these multiple transmission requests uses the fixed priority scheme given in Table 6-10.

Table 6-10. Priority of Transmission Requests

Priority	Register
Highest	Currently Transmitting Register
•	XCVR0
•	XCVR1
•	XCVR2
•	XCVR9
•	XCVRA
Lowest	XCVRB

### 6.4.3 Broadcast Transmit Mode

In *broadcast transmit* mode a device transmits a broadcast message and determines if the message was received. The broadcast message is characterized by its function code and can be received by any device with an XCVRn register that is set up to receive a message with that same function code. The broadcast transmit mode requires SMSG = 0 and RXEN = 0, as shown in Table 6-8.

The broadcast transmit mode has two states and two transitions, which are characterized by the values of certain bits in the XCVRn register. Figure 6-6 shows the states and transitions, and Table 6-11 shows the associated bit activity in the XCVRn register.

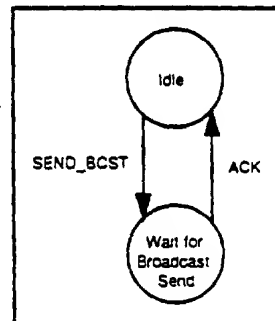


Figure 6-6. Broadcast Transmit State Diagram

Table 6-11. Broadcast Transmission Bit Activity

State/Transition	Associated Register Bits
Idle state	TXIPR = 0
Wait for Broadcast Send state	TXIPR = 1
SEND_BCST transition	Software writes TXIPR = 1
ACK transition	Hardware drives TXIPR = 0

The following example illustrates a typical programming sequence for transmitting a broadcast message.

1. Configure the register and, ~~for~~ this example, set the TXINT bit to enable its corresponding meta-interrupt.

**XCVRn**

Bit No.	31	30	29	28	27	26	25	24
Name	MSG	RXEN	BINT2	RSVD	CINT	BINT	NINT	TXINT
Value	0	0	0	0	0	0	0	1

(Note: The choice of which (if any) meta-interrupts to enable depends on the application. Setting the TXINT bit, which enables a meta-interrupt to be generated when hardware clears the TXIPR bit, is just an illustrative example. Instead, you might set BINT or BINT2, or you could choose to enable none of the meta-interrupts.

2. Write the FCODE and DATA fields into bits 15-0.
3. Initiate the transmit operation by writing bits 23-16 to set TXIPR:

**XCVR<sub>n</sub>**

Bit No.	23-22	21	20	19	18	17	16
Name	RSVD	EXTRA	SUCCESS	COMPLETE	FMATCH	NACK	TXIPR
Value	0	0	0	0	0	0	1

4. Wait for a meta-interrupt:

If TXIPR = 0, the transmission is complete. (Hardware clearing this bit generates the TXINT meta interrupt.) If TXIPR = 1, the meta-interrupt was not generated by this register.

A new broadcast transmission can then begin.

The following notes may preclude confusion regarding the operation of the NACK bit and the FMATCH bit.

1. Hardware may toggle the NACK bit in broadcast mode. However, the NACK bit has no significance in this mode and can be ignored.
2. When the 82750PD is in broadcast transmit and broadcasts a message with a certain function code, hardware sets the FMATCH bit in **all** transceiver registers having the same function code — including the register that broadcasts the message.

#### 6.4.4 Broadcast Receive Mode

The 82750PD can receive different types of broadcast messages, where each type is characterized by a unique function code. For each type of message to be received there must be an XCVR<sub>n</sub> register set up with the corresponding function code. When a message with a function code X is received, the data field bits in the packet are latched into the data field of the XCVR<sub>n</sub> register

that has X in the FCODE field. (The 82750PD ignores messages with function codes not in its range.) The XCVRn register that receives the message with function code X can be set up to trigger a meta-interrupt.

The following example illustrates a typical programming sequence to receive broadcast messages.

1. Configure the XCVRn register to listen for broadcast messages by writing bits 31-24:

XCVRn

Bit No.	31	30	29	28	27	26	25	24
Name	SMMSG	RXEN	BINT2	RSVD	CINT	BINT	NINT	TXINT
Value	0	1	0	0	0	0	0	0

2. Write the FCODE bits (bits 15-12) with the function code for the type of message to be received. (When the 82750PD writes to the register in this mode, the bits in the DATA field (bits 11-0) are "don't care"; they are not written to the register.)

(Writes to the FCODE field take effect in one 82750PD core clock. Incoming messages are received, synchronized to the 82750PD core clock, and presented to this XCVRn register. If this register is written in the exact clock that a message is presented, then the "old" contents of the register are used.)

3. Clear the FMATCH bit by writing to bits 23-16:

XCVRn

Bit No.	23	22	21	20	19	18	17	16
Name	RSVD	RSVD	EXTRA	SUCCESS	COMPLETE	FMATCH	NACK	TXIPR
Value	0	0	0	0	0	0	0	0



Note that the FMATCH bit must always be cleared when a new function code is written in broadcast receive mode. Otherwise, a message with the new function code goes undetected.

4. Enable the desired broadcast meta-interrupt (BINT or BINT2) by writing bits 31-24 (BINT is enabled in this example.).

XCVR<sub>n</sub>

Bit No.	31	30	29	28	27	26	25	24
Name	MSG	RXEN	BINT2	RSVD	CINT	BINT	NINT	TXINT
Value	0	1	0	0	0	1	0	0

When a broadcast message with the selected function code is received, the FMATCH bit is set, the DATA field is updated with the data from the packet, and a meta-interrupt is generated. At this point, the device should read the DATA field of the XCVR<sub>n</sub> register. If a second message of the same type arrives, it overwrites the DATA field.

The DATA field always contains the data from the most recently received message (with the specified FCODE). Software can determine that this register received a message by reading the FMATCH bit. Software need not clear the FMATCH bit for the receipt of a new message (with the same FCODE) to generate another meta-interrupt. However, to identify uniquely each message received by this register, software must clear the FMATCH bit after each message is received and before the next message is received.

#### 6.4.5 Sending a Request and Receiving a Completion

When the 82750PD attempts to send a service request, there are two possible outcomes:

- The message is sent successfully (accepted by the target), and the 82750PD receives a service completion message.
- The message is not acknowledged (NACKed).

Either one of these outcomes occurs within approximately 21  $\mu$ s. The hardware does not provide an automatic resend capability. It is software's responsibility to repeat the attempt to send. The hardware provides a one-shot attempt to send a message; it is either sent or is NACKed.

Service completion messages are a special class of broadcast transmit messages. After a device has received a service request and has completed the task, it sends a service completion message back to the device that requested the service. These messages are asynchronous and cannot be NACKed; they must be accepted. Each service completion message is paired with a service request message. The XCVR<sub>n</sub> register that sends the service request receives the service completion. As a result, the number of outstanding service requests is then limited to the number of registers configured for service completion (a maximum of 12 XCVR<sub>n</sub> registers with SMSG = 1 and RXEN = 0). Figure 6-7 shows a state diagram for sending a service request and receiving a service completion, and Table 6-12 lists the bit activity in the XCVR<sub>n</sub> register.

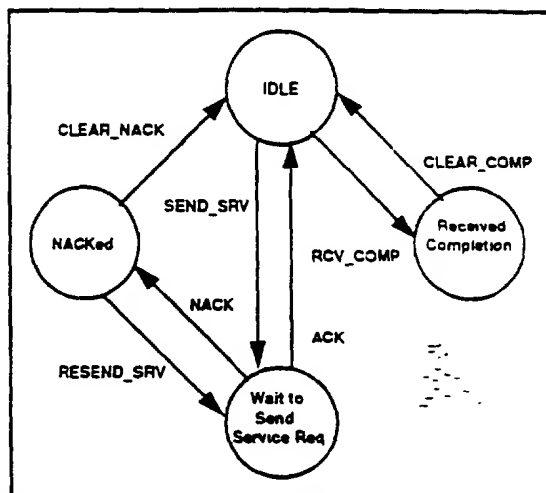


Figure 6-7. Send Service Request State Diagram

Table 6-12. Send Service Request Bit Activity

State/Transition	Bit Values in XCVR <sub>n</sub>
State: Idle	TXIPR = 0
State: wait to send service request	TXIPR = 1, NACK = 0, COMPLETE = 0
State: NACKed	TXIPR = 0, NACK = 1, COMPLETE = 0
State: received service completion	TXIPR = 0, NACK = 0, COMPLETE = 1
Transition: SEND_SRV	Software writes TXIPR = 1, NACK = 0, COMPLETE = 0
Transition: ACK	Hardware drives TXIPR = 0
Transition: NACK	Hardware drives TXIPR = 0, NACK = 1
Transition: RESEND_SRV	Software writes TXIPR = 1, NACK = 0, COMPLETE = 0
Transition: RCV_COMP	Hardware drives COMPLETE = 1
Transition: CLEAR_NACK	Software writes TXIPR = 0, NACK = 0, COMPLETE = 0
Transition: CLEAR_COMP	Software writes TXIPR = 0, NACK = 0, COMPLETE = 0

**NOTE**

*The register returns to the IDLE state after the service request has been sent. It is software's responsibility to "remember" that a service request was sent and that a corresponding service completion is expected.*

The following example illustrates a typical programming sequence to transmit a service request message and receive the service completion message.

1. Configure the XCVRn register to transmit a service request and receive a service completion by writing bits 31-24:

**XCVRn**

Bit No.	31	30	29	28	27	26	25	24
Name	MSG	RXEN	BINT2	RSVD	CINT	BINT	NINT	TXINT
Value	1	0	0	0	0	0	0	0

2. Write the target ID and service number to the DATA field (bits 11-2). DATA field bits 1 and 0 may be written with any value. Write 0Eh (the service request code) to the FCODE field (bits 15-12).
3. Enable the service completion (CINT) and the NACK (NINT) meta-interrupts by writing bits 31-24:

**XCVRn**

Bit No.	31	30	29	28	27	26	25	24
Name	MSG	RXEN	BINT2	RSVD	CINT	BINT	NINT	TXINT
Value	1	0	0	0	1	0	1	0

4. Initiate the transmission by writing to bits 23-16 to clear the NACK, COMPLETE, and SUCCESS bits and to set the TXIPR bit:

**XCVRn**

Bit No.	23	22	21	20	19	18	17	16
Name	RSVD	RSVD	EXTRA	SUCCESS	COMPLETE	FMATCH	NACK	TXIPR
Value	0	0	0	0	0	0	0	0

5. Wait for a meta-interrupt. The pseudo code that follows describes the meanings of the bit values and the actions to be taken.

if TXIPR = 1 then the interrupt was not generated by this register

else if NACK = 1 then go to step 4 (must retry because the target was unable to accept the command)

else if SUCCESS = 1  
then the service was completed successfully  
else service was not completed successfully.

Another attempt to send a service request and receive a service completion (using the same target ID and service number) can be executed by simply going to step 4.

#### 6.4.6 Receiving a Request and Sending a Completion

The 82750PD can receive service request messages from other devices. An incoming service request is held in the Receive Service Register (RSRV). The 82750PD reads this register, performs the requested service, and sends a service completion message to the originator of the service request. The following subsections describe the RSRV register and the operations for receiving a request and sending a completion.

Receive Service Register (RSRV)

The Receive Service Register (RSRV) holds a service request message that arrives from another device. The 82750PD reads the RSRV register to find out what service it is asked to perform. The 82750PD must read the RSRV register to access the first service request before the RSRV register can accept a second service request. If a second service request arrives before the 82750PD reads the RSRV register for the first service request, the second service request is NACKed. Table 6-13 shows the RSRV register.

Table 6-13. Receive Service Register

Mnemonic: RSRV

Address: 00FF0034-00FF0037

Access: See bit descriptions.

Reset State: See bit descriptions.

Bit No.	31-25	24	23-17	16	15-12	11-2	1-0
Name	RSVD	IINT	RSVD	VALID	FCODE	SERV_NUM	RSVD

**SERV\_NUM** (Service Number, bits 11-2: R/O, uninitialized by reset). This field latches the service number (packet bits 10-13 and 15-20) from the received message.

**FCODE** (Function Code, bits 15-12: R/O; uninitialized by reset). This 4-bit field always contains 0Eh (the service request code) after a service request is received. These bits are uninitialized by reset.

**VALID** (Valid, bit 16: R/W; reset state = 0). This can be used by 82750PD core microcode or the host to determine if a completion message was received. The SynchroLink hardware sets this bit when a service completion message is received and acknowledged. An incoming completion message is defined as a message satisfying these criteria:

- The function code (bits 6-9) = 0Eh (service request).
- The service number (packet bits 10-13, 15-20) is a *recognized service number* (see the Match Service register (MSRV) described in Section 6.4.8).

If the message does not meet these criteria, the VALID bit is not set and the service acknowledge bit (bit 22) in the message packet is not set.

The 82750PD core microcode and the host can reset the valid bit, but writing a '1' has no effect. If software clears this bit and hardware sets this bit in the same clock, the bit is cleared.

**IINT** (Incoming Message Interrupt, bit 24: R/W; reset state = 0). Setting this bit enables a NOT-VALID to VALID transition (bit 6) to generate the V1CODE VBUS code.

### Operations

---

This section describes procedures for receiving a service request and sending the corresponding service completion message.

Assume that the 82750PD uses one of its transceiver registers (denoted by XCVRX) to send a service request message to the 82750PD. The 82750PD reads the service request from its RSRV register, performs the service, and sends a service completion message to the 82750PD. The 82750PD sends the service completion whether it was successful or unsuccessful in performing the requested service. Note that the 82750PD hardware does not send the service completion. The 82750PD **software** is responsible for sending the service completion.

When the 82750PD receives the service completion message, the COMPLETE bit is set in the XCVRX register. If the service was performed successfully, the SUCCESS bit is also set in the XCVRX register.

The following example illustrates a typical programming sequence where the 82750PD receives a service request and sends the service completion. Before beginning this sequence, the 82750PD should configure a transceiver register (denoted by XCVRX) for a broadcast transmit operation (see Section 6.5.3).

1. Clear the VALID bit in the RSRV register by writing 00h to bits 23-16.
2. Enable the incoming message interrupt (IINT) in the RSRV register by writing 01h to bits 31-24.
3. Wait for a meta-interrupt. When it arrives, check the VALID bit. If the VALID bit is set, a service request message has arrived. Otherwise, the meta interrupt was not caused by a service request message.  
(For the following steps, assume that the VALID bit is set.)
4. Read the FCODE and DATA bits 15-0 from the RSRV register. (The FCODE should be 0Eh, the code for a service request.)
5. Perform the service request as indicated in the DATA field.
6. Initiate the following *broadcast transmit* operation to send the service completion message. Register XCVRX is configured for broadcast transmit.
  - a) Write 00h to bits 31-24 of the XCVRX register.
  - b) Write the FCODE and DATA fields in bits 15-0 of the XCVRX as follows:



Load the FCODE field (bits 15-12) with 0Fh (the service completion FCODE).

Load DATA bits 11-2 with the 10-bit DATA field read from the RSRV register.

(When the data returned to the 82750PD is the same as the data that the 82750PD sent, hardware sets the COMPLETE bit in the 82750PD's transceiver register.)

If the service was completed successfully, set bit 0. (This causes the SUCCESS bit to be set in the 82750PD's XCVRX register.)

c) Set the TXIPR bit in the XCVRX register by writing 01h to bits 23-16.

d) Poll the TXIPR bit until it is cleared by hardware. This indicates that the transmission is complete.

7. Clear the VALID bit in the RSRV register by writing 00h to bits 23-16. This allows the hardware to accept another service request. As long as the VALID bit is set, incoming service requests are NACKed.

At this point the 82750PD hardware is prepared to accept another service request (i.e., prepared to recommence at step 3.)

#### 6.4.7 Message Status Register (MSTATUS)

The Message Status Register, shown in Table 6-14, is a read-only register that provides status information on the XCVRn registers and the RSRV register.

Table 6-14. Message Status Register

Mnemonic: MSTATUS

Address: 00FF0030h-00FF0033

Access: R/O

Reset State: 00h

Bit No.	31-25	24	23-22	21-20	19-18	17-16
Name	0	RSRV STAT	XCVRB STAT	XCVRA STAT	XCVR9 STAT	XCVR8 STAT

Bit No.	15-14	13-12	11-10	9-8	7-6	5-4	3-2	1-0
Name	XCVR7 STAT	XCVR6 STAT	XCVR5 STAT	XCVR4 STAT	XCVR3 STAT	XCVR2 STAT	XCVR1 STAT	XCVR0 STAT

**XCVRn STAT (Transceiver Status).** This two-bit field reflects the status of an XCVRn register (n = 0, 1, ... 9, A, B). The definitions of these bits depend on the state of the SMSG bit in that register, as shown in Table 6-15.

Table 6-15. Transceiver Status Bits and the SMSG Bit.

SMSG	XCVRn STAT[1] in the MSTATUS Register	XCVRn STAT[0] in the MSTATUS Register
0	FMATCH	TXIPR
1	NACK OR* COMPLETE	TXIPR

\* Logical inclusive OR

Software can monitor bit 0 of the XCVRn STAT field to determine if the XCVRn register has completed a transmission. In a broadcast receive situation (SMSG = 0), software can monitor bit 1 to determine if the XCVRn register has received a message with a function code that matches the function code in XCVRn. When XCVRn is waiting for a response to a service request, software can monitor bit 1 to determine if any response has been received i.e., the request has been NACKed or the service has been completed.

## SynchroLink

**RSRV STAT** (Receive Service Status, bit 24). This bit reflects the state of the valid bit the RSRV register.

### 6.4.8 Match Service Register (MSRV)

When the 82750PD sees a service request message on the SynchroLink, it checks to see if the service number falls within the range assigned to the 82750PD. If it does, this is a *recognized service number*, and the 82750PD responds to the request. (A paragraph below defines a recognized service number.) If the service number is not recognized, the 82750PD can ignore the service request. The Match Service Register (see Table 6-7) defines the range of service numbers that the 82750PD recognizes. Software must not alter this register unless MSG\_EN = 0 and DMSG\_EN = 0 in the Message Configuration Register (defined below). Such an alteration may result in incorrect message reception.

Table 6-16. Match Service Register

Mnemonic: MSRV

Address: 00FF0038h-00FF003B

Access: R/W

Reset State: See bit descriptions.

Bit No.	31-25	27-18	17-16	15-12	11-2	1-0
Name	0	SRV_MASK	0	0	SRV_DATA	0

**SRV\_DATA** (Service Data, bits 11-2: uninitialized by reset). The bits in this field are compared with the service number bits that are selected by the **SRV\_MASK** field. If the bits match, the received message is recognized, and the 82750PD responds to the request.

A recognized service number is defined as a service number that satisfies the following criterion:

(packet bit[10] = SRV\_DATA[9]) OR (SRV\_MASK[9]=0) AND  
 (packet bit[11] = SRV\_DATA[8]) OR (SRV\_MASK[8]=0) AND  
 (packet bit[12] = SRV\_DATA[7]) OR (SRV\_MASK[7]=0) AND  
 (packet bit[13] = SRV\_DATA[6]) OR (SRV\_MASK[6]=0) AND  
 (packet bit[15] = SRV\_DATA[5]) OR (SRV\_MASK[5]=0) AND  
 (packet bit[16] = SRV\_DATA[4]) OR (SRV\_MASK[4]=0) AND  
 (packet bit[17] = SRV\_DATA[3]) OR (SRV\_MASK[3]=0) AND  
 (packet bit[18] = SRV\_DATA[2]) OR (SRV\_MASK[2]=0) AND  
 (packet bit[19] = SRV\_DATA[1]) OR (SRV\_MASK[1]=0) AND  
 (packet bit[20] = SRV\_DATA[0]) OR (SRV\_MASK[0]=0)

**SRV\_MASK** (Service Mask, bits 27-18: uninitialized by reset). This field selects the bits of the service number that are to be compared with the bits in the **SRV\_DATA** field. If bit x in the **SRV\_MASK** field is set, then bit x in the service number is compared to bit x in the **SRV\_DATA** field.

#### 6.4.9 Configuring the SynchroLink Interface

The SynchroLink interface and the internal VBUS are configured by writing to the MCFG register. The following subsections describe the MCFG register and give an example of a configuration procedure.

##### *Message Configuration Register (MCFG)*

Table 6-17 shows the Message Configuration Register (MCFG).

Table 6-17. Message Configuration Register

Mnemonic: MCFG

Address: 00FF003C-00FF003F

Access: R/W

Reset State: See bit descriptions.

Bit No.	31-28	27-24	23	22-17	16
Name	V2CODE	V1CODE	PROTECT	0	EXTVBUS

Bit No.	15-12	11	10	9	8	7-6	5-4	3	2-0
Name	RSVD	CLK_EN	ARB_EN	DMSG_EN	MSG_EN	RSVD	CLKDIV	RSVD	MSG_ID

**MSG\_ID** (Message ID, bits 2-0: uninitialized by reset). This field defines the ID of the 82750PD in the SynchroLink protocol.

**CLKDIV** (Clock Division, bits 5-4: reset state = 0) When the 82750PD is the arbiter (ARB\_EN = 1), these bits define the SynchroLink clock frequency (SLCLK) in terms of the SFB1 clock frequency (MCLK) as listed in Table 6-18.

Table 6-18. SynchroLink Clock Frequency.

CLKDIV Bits		SynchroLink Frequency (SLCLK)
Bit 5	Bit 4	
0	0	MCLK/2
0	1	MCLK/4
1	0	MCLK/6
1	1	MCLK/8

**MSG\_EN** (Message Enable, bit 8: reset state = 0). Setting this bit enables the 82750PD to send and receive messages. Clearing MSG\_EN prevents the 82750PD from sending or receiving messages. However, it does continue to monitor the SynchroLink to identify message start bits. The effect of changes in the MSG\_EN bit do not take

effect until a break (ten consecutive '0' bits) on the SynchroLink is detected. This prevents a change from occurring in mid-message. Software should use the DMSG\_EN bit to determine when the 82750PD responds to changes in the MSG\_EN bit.

**DMSG\_EN** (Delay Message Enable, bit 9; reset state = 0) The value of this R/O bit follows the value written to the MSG\_EN bit. However, DMSG\_EN changes state only when the actual status of the 82750PD changes. (The actual change in status is delayed until the next break (ten consecutive '0' bits) on the two wire bus after a change in the MSG\_EN bit.) By writing the MSG\_EN bit and then polling the DMSG\_EN bit software can determine when the change has occurred.

**ARB\_EN** (Arbiter Enable, bit 10; reset state = 0). Setting this bit defines the 82750PD as the arbiter for the SynchroLink, i.e., the 82750PD polls the devices for requests to use the SynchroLink. For ARB\_EN = 0, the 82750PD does not poll for requests from other devices.

**CLK\_EN** (Clock Enable, bit 11; reset state = 0). Setting this bit causes the 82750PD to drive the SynchroLink clock at a frequency specified by the CLKDIV bits (bits 5-4).

**EXTVBUS** (External VBUS, bit 16; reset state = 0). Setting this bit enables the 82750PD VBUS to be driven from the external VBUSIN pins of the chip. Do not select this mode unless the host bus type is ISA or PCI. Furthermore, do not select this mode if you want to allow meta-interrupts to the 82750PD core. If the EXTVBUS bit is set, the VBUS codes generated by the SynchroLink interface are not driven to the 82750PD core. Instead, the states of the external VBUSIN pins are driven to the core via the VBUS.

**PROTECT** (Protect, bit 23: reset state = 0). Setting/clearing this bit disables/enables further writes to this register. During normal operation, this bit can be cleared only by writing bits 31-24 with data that matches the data already in bits 31-24. Reading this bit always returns '0', even if the bit is set.

**V1CODE** (bits 27-24: uninitialized by reset). These bits define the code to be driven onto the 82750PD core's VBUS in response to a meta-interrupt in the V1CODE group (see Section 6.5).

**V2CODE** (bits 31-28: uninitialized by reset). These bits define the code to be driven onto the 82750PD core VBUS in response to a meta-interrupt in the V2CODE group (see Section 6.5).

### Configuration Procedure

---

The following example illustrates a typical programming sequence for configuring the SynchroLink interface. Software must confirm that no transmission requests are pending on the SynchroLink (i.e., TXIPR = 0 in all transceiver registers) and that all expected completion messages have been received.

1. Clear the MSG\_EN bit without changing the states of the other bits in byte 1 (bits 15-8). (If the 82750PD is the arbiter, and the CLK\_EN and ARB\_EN bits are cleared, a break never occurs on the SynchroLink, and the DMSG\_EN bit is never cleared. If another device is supplying the clock signal, the CLK\_EN bit can be clear.)
2. Poll the DMSG\_EN bit until it is clear.
3. Examine all registers for any "last minute" messages that might have been received.

4. Assign the 82750PD its message ID by writing the MSG\_ID bits.
5. Configure the VBUS codes by writing the V1CODE and V2CODE bits.
6. Specify an internal or external VBUS source by writing the EXTVBUS bit.
7. Select the desired transceiver register modes by writing the SMSG and RXEN bits in **all** transceiver registers (XCVRn, n = 01h - 0Bh).
8. Enable the 82750PD to send **and** receive messages by setting the MSG\_EN bit **without changing** the states of the other bits in byte 1 of the register.
9. Poll the DMSG\_EN bit. When it is set, the SynchroLink interface is configured.

## 6.5 Meta-Interrupts

The 82750PD has an interrupt system composed of interrupts and meta-interrupts. The interrupts are described in Section 3.14. This section covers meta-interrupts. The combined system of interrupts and meta-interrupts is described in Section 5.7.

Some events that originate from devices on the SFBI are of special interest to the 82750PD. For example, the 82750PD should be informed when the capture device has a frame that is ready for compression. Such events alter registers in the SynchroLink interface. Although the 82750PD software could poll these registers to detect the event, a meta-interrupt can signal the 82750PD core directly (via the VBUS) when the event occurs.



*Meta-interrupts* are a mechanism for notifying the 82750PD core of certain events associated with devices on the SFB<sub>I</sub> without causing an 82750PD interrupt to the host. When an *SFB<sub>I</sub> event* occurs, it sets or clears a register bit in the SynchroLink interface. At the programmers discretion, that change in the register causes a signal to be sent over the VBUS to the core, where the signal alters the core registers *fcnt* and/or *lcnt*. To discover the occurrence of a meta-interrupt, the core microcode must poll these registers.

Table 6-19 lists the SFB<sub>I</sub> events (i.e., events that can be programmed to generate meta-interrupts to the 82750PD core), the effect of each event on registers in the SynchroLink interface, and the register bit that enables the meta-interrupt.

Table 6-19. Enabling Meta-Interrupt Events to Generate VBUS Codes

SFBI Event	Effect on Event SynchroLink Interface Registers	Enabling Bit
The 82750PD has received and acknowledged an incoming service request message	Hardware sets VALID (bit 16) in RSRV register	INT (bit 24) in RSRV register
A message transmission is complete	Hardware clears TXIPF (bit 16) in XCVR <sub>n</sub> register	TXINT (bit 24) in XCVR <sub>n</sub> register
A service request from the 82750PD was not accepted by the target device	Hardware sets NACK (bit 17) in XCVR <sub>n</sub> register	INT (bit 25) in XCVR <sub>n</sub> register
A device that was sent a service request by the 82750PD has replied	Hardware sets COMPLETE (bit 19) in XCVR <sub>n</sub> register	CINT (bit 27) in XCVR <sub>n</sub> register
The 82750PD has received a broadcast message with a certain function code *	Hardware sets FMATCH (bit 18) in XCVR <sub>n</sub> register	BINT (bit 26) in XCVR <sub>n</sub> register
The 82750PD has received a broadcast message with a certain function code *	Hardware sets FMATCH (bit 18) in XCVR <sub>n</sub> register	BINT2 (bit 29) in XCVR <sub>n</sub> register

\* These events are identical, but the meta-interrupts are enabled by different bits and generate different VBUS codes

## Appendix A 82750PD Registers

This appendix contains tables of the 82750PD registers. Information in the tables includes the register name, its mnemonic, the address, the read/write access, the reset value, and the bit descriptions. The registers are listed in the alphabetical order of the mnemonics.

82750PD core registers that consist of a single field have been excluded. The mnemonics for these registers are listed in Table A-1.

Table A-1. 82750PD Core Registers Excluded from Appendix A

Core Registers	
alu	*outN: N = 1,2
cnt cnt2	pc
dramN-- N=1-4	pxnnt
dramN-- N=1-4	rn: N = 0-15
*dramN: N=1-4	shift
lcnt	shift-r, l, d
inN-hi inN-lo N = 1,2	stat-c
*inN: N = 1,2	stat-hi stat-lo
lcnt	stat-ram
outN--	*stat
outN-hi outN-lo N = 1,2	*stat#

## 82750PD Registers

Table A-2. EISA POS3 Register

Mnemonic: BASE

Address: 0zCB8Eh\*

Access: W/O

Reset State: 75h\*\*

Bit No.	0
Name	IOB[7:0]

Bit No.	Name	Description
7-0	IOB[7:0]	IOB[7:0] = I/O Base[7:0]***

\* z in the Address field denotes slot-dependent bits.

\*\* If IO\_BASE[8] = 1, the reset state corresponds to IO\_START = x2EAh where x depends on the Device ID (see Section 5.3)

\*\*\* I/O Base[8] is latched from straps or jumpers at reset.

Table A-3. EISA POS2 Register

Mnemonic: BCNTL

Address: 0zC84h\*

Access: W/O

Reset State: Not available.

Bit No.	7-1	0
Name	RSVD	ENABLE

Bit No.	Name	Description
0	ENABLE	ENABLE = IO_ENABLE = MEM_ENABLE. Clearing this bit disables the host I/O registers and disables the "EMS" window even if the EWE bit (bit 0 in the CFG 06 register) is set.

\* z in the Address field denotes slot-dependent bits.

Table A-4. ALU Condition Code Register

Mnemonic:

Address\*

Access: R/W

Reset State: Not available

Bit No.	15-8	7	6	5	4	3	2	1	0
Name	RSVD	r0 MSB	r0 LSB	Loop Counter Zero	ALU Zero	ALU Sign	ALU Overflow	ALU Carry Out	0

Bit No.	Name	Description
0	0	(R/O) This bit reads as '0' and is unaffected by writes
1	ALU Carry Out	ALU Carry Out (R/W) The value of this bit equals the carry out of the most significant bit position in the ALU. This bit is '0' for all logical operations.
2	ALU Overflow	ALU Overflow (R/W). The value of this bit is (ALU carry out) XOR (Carry in to most significant bit of ALU result). This bit is '0' for all logical operations.
3	ALU Sign	ALU Sign (R/W) The value of this bit is the most significant bit of the ALU result.
4	ALU Zero	ALU Zero (R/W) This bit is set only if all bits of the ALU result are '0'.
5	Loop Counter Zero	Loop Counter Zero (R/O). This bit is set only if the value of the loop counter is '0'.
6	r0 LSB	r0 LSB (R/O) The value of this bit is the value of the least significant bit of register r0.
7	r0 MSB	r0 MSB (R/O) The value of this bit is the value of the most significant bit of register r0.
15-8	RSVD	

\* See A and B bus addresses

Table A-5. Core Control Register

Mnemonic: *ccontrol*

Byte Offset: 100h

Access: W/O

Reset State: 01h

Bit No.	15	14	13	12	11-8
Name	CORE_EN	RSVD	PMON/FRZ	DSYNC	RSVD

Bit No.	7	6	5	4	3	2	1	0
Name	EFL_M	OFL_M	RSVD	RSVD	VBI_M	MCINT_M	STEP	HALT

Bit No.	Name	Description															
0	HALT	<p><b>Halt.</b> For CORE_EN = 1, this bit and the STEP bit place the core in normal run mode, single step mode, or the Halt state, as shown below:</p> <table> <tr> <td>STEP</td><td>HALT</td><td></td></tr> <tr> <td>0</td><td>0</td><td>Normal Run.</td></tr> <tr> <td>0</td><td>1</td><td>Halt</td></tr> <tr> <td>1</td><td>0</td><td>Normal Run</td></tr> <tr> <td>1</td><td>1</td><td>Single Step (Core executes one instruction and halts).</td></tr> </table>	STEP	HALT		0	0	Normal Run.	0	1	Halt	1	0	Normal Run	1	1	Single Step (Core executes one instruction and halts).
STEP	HALT																
0	0	Normal Run.															
0	1	Halt															
1	0	Normal Run															
1	1	Single Step (Core executes one instruction and halts).															
1	STEP	<b>Step.</b> For CORE_EN = 1, this bit and the HALT bit control the running mode for the core. See the HALT bit above.															
2	MCINT_E	<b>Microcode Interrupt Enable.</b> Setting this bit enables the microcode interrupt condition to interrupt the core.															
3	VBI_E	<b>Vertical Blanking Interrupt Enable.</b> Setting this bit enables the vertical blanking interrupt condition to interrupt the core.															
4-5	RSVD	Reserved															
6	OFL_E	<b>Odd Field Interrupt Enable.</b> Setting this bit enables the odd field interrupt condition to interrupt the core.															
7	EFL_E	<b>Even Field Interrupt Enable.</b> Setting this bit enables the even field interrupt condition to interrupt the core.															
11-8	RSVD	Reserved															
12	DSYNC	<b>Disable Synchronizers.</b> Setting this bit disables the synchronizers for the HREQ#/HALEN# signal.															
13	PMON/FRZ	<p><b>Performance Monitor/Freeze.</b> This bit determines which output signal is on the PMFRZ# pin. This bit is functional only for ISA and PCI buses.</p> <p>1 = Output signal FRZ# is on PMFRZ# pin 0 = Output signal PMON# is on PMFRZ# pin</p>															

Table A-5. Core Control Register (Continued)

14	RSVD	Reserved
15	CORE_EN	<b>Core Enable.</b> This bit must be set for the core to operate. It must be set for the core to be in single-step mode or in the Halt state as well for normal running mode. See the HALT and STEP bits.

Table A-6. I/O Base CFG Register

Mnemonic: CFGBASE

Address: CFG 01

Access: R/O

Reset State: See bit definitions

Bit No.	7-0
Name	IOB[7:0]

Bit No.	Name	Description
7-0	IOB[7:0]	These bits indicate the state of the IO BASE 'POS' field. See Section 5.2, "Host Interface Address Configuration" for how these bits are reset and/or changed.

Table A-7. Bus CFG Register

Mnemonic: CFGBUS

Address: CFG 02

Access: R/W

Reset State: 00h

Bit No.	7-0
Name	BCFG[7:0]

Bit No.	Name	Description
7-0	BCFG[7:0]	These configuration bits depend on the bus type.

Table A-8. CFG Register Data Register

Mnemonic: CFGDATA

Address: REG6-BYTE1

Access: R/W

Reset State\*

Bit No.	31-0
Name	—

\*This is not a physical register

Bit No.	Name	Description
15-0		This register holds the data that is read from or written to a configuration register that is accessed via the CFGNUM register (REG6-BYTE0).

Table A-9. General Configuration Register

Mnemonic:

Address: CFG 00

Access: R/O

Reset State\*

Bit No.	7	6	5-3	2-0
Name	MTYP	RSVD	BTYP[5:3]	DID[2:0]

\* Determined from strapping options at reset

Bit No.	Name	Description
2-0	DID[2:0]	<b>Device ID.</b> These bits define the device ID of the 82750PD. The bit values are determined from the configuration straps at reset.
5-3	BTYP[2:0]	<b>Bus Type.</b> These bits define the bus type. The bit values are determined from the configuration straps at reset.
6	RSVD	
7	MTYP	<b>Memory Type.</b> This bit is set if 4 Mb DRAM chips are used in the SFB. The bit is latched from a strap at reset.*



Table A-10. INT CFG Register

Mnemonic: CFGINT

Address: CFG 03

Access: See bit descriptions.

Reset State: See bit descriptions.

Bit No.	7	6	5	4	3	2	1	0
Name	MEMEN	IOEN	IOB8	HALT	TEST	VIP	VIE	VIT

Bit No.	Name	Description
0	MEMEN	This R/O bit reflects the state of the MEM_ENABLE 'POS' field (See Section 5.2 for a description of this bit.)
1	IOEN	This R/O bit reflects the state of the IO_ENABLE 'POS' field (See Section 5.2 for a description of this bit.)
2	IOB8	This R/O bit indicates the state of the 8th bit of the IO_BASE 'POS' field. This bit is latched from a strap during reset.
3	HALT	This R/W bit is used to control the 82750PD core. Setting HALT causes the PB core execution to halt. Reset state = 0.
4	RSVD	
5	VIP	This R/W bit is the interrupt level polarity bit for 82750PB interrupts. If VIP is set, the interrupt is active high. If VIP is clear, the interrupt is active low. Reset state = 0.
6	VIE	This R/W bit is the interrupt level enable bit for 82750PB interrupts. If VIE is set, the PB interrupt is presented on the IRQ pin. If VIE is clear, the PB interrupt is three-stated. Reset state = 0.
7	VIT	This R/W bit is the interrupt level type bit for 82750PB interrupts. If VIT is set, the PB interrupt is totem pole output. If VIT is clear, the PB interrupt is open drain output. Reset state = 0.

Table A-11. CFG Register Number Register

Mnemonic: CFGNUM

Address: REG6-BYTE0

Access: R/W

Reset State: 00h

Bit No.	7-4	3-0
Name	RSVD	R[3:0]

Bit No.	Name	Description
0	R0	Bit 0 (LSB) of the CFG register number
1	R1	Bit 1 of the CFG register number
2	R2	Bit 2 of the CFG register number
3	R3	Bit 3 (MSB) of the CFG register number
4-7	RSVD	

Table A-12. SFBI CFG 04 Register

Mnemonic: CFGSFBI4

Address: CFG 04

Access: R/W

Reset State: 00h

Bit No.	7	6	5	4	3	2	1	0
Name	PAGE	REFEN	HCAS	HLCH	RSVD	MSZ2	MSZ1	MSZ0

Bit No	Name	Description												
2-0	MSIZ[2-0]	<p><b>Memory Size.</b> This 3-bit R/W field defines the size of the Shared Frame Buffer (SFB)</p> <table><tr><th>MSIZ[2-0]</th><th>SFB Size</th></tr><tr><td>0 1 0</td><td>1MB (also configures SFBC as 32-bit)</td></tr><tr><td>0 1 1</td><td>2MB</td></tr><tr><td>1 0 0</td><td>4MB</td></tr><tr><td>1 1 0</td><td>6MB</td></tr><tr><td>1 1 1</td><td>8MB</td></tr></table> <p><i>Note: All other values are illegal. (Illegal values produce memory cycle and "readys" but the data is not guaranteed)</i></p>	MSIZ[2-0]	SFB Size	0 1 0	1MB (also configures SFBC as 32-bit)	0 1 1	2MB	1 0 0	4MB	1 1 0	6MB	1 1 1	8MB
MSIZ[2-0]	SFB Size													
0 1 0	1MB (also configures SFBC as 32-bit)													
0 1 1	2MB													
1 0 0	4MB													
1 1 0	6MB													
1 1 1	8MB													
3	RSVD													
4	HLCH	<b>Half Latch.</b> If this R/W bit is set, the data latch is closed one-half clock early												
5	HCAS	<b>Half CAS.</b> If this R/W bit is set, the CAS signal is activated one-half clock early												
6	REFEN	<b>Refresh Enable.</b> Setting this R/W bit enables the SFB interface to cause a CAS-before-RAS refresh cycle in response to 82750PD refresh cycles												
7	PAGE	<b>Page.</b> This bit determines whether the 82750PD leaves the SFB page open or closes it while the 82750PD owns the SFB. Setting this bit leaves the SFB page open (RAS low). Clearing this bit closes the SFB page (RAS high)												

# 82750PD Registers

Table A-13. SFBI CFG 05 Register

Mnemonic: CFGSFBI5

Address: CFG 05

Access: R/W

Reset State: 00h

Bit No.	7	6	5	4	3	2	1	0
Name	MMPR1	MMPR0	QUASI	CCNT4	CCNT3	CCNT2	CCNT1	CCNT0

Bit No.	Name	Description
4-0	CCNT[4-0]	These 5-bit R/W field that defines the maximum number of SFBC clocks (0 - 32) that the 82750PD is allowed to issue before giving up the SFBC in response to GRANT being deasserted. If GRANT is deasserted, the 82750PD continues using the SFBC (if it has any outstanding requests) until it has issued the maximum number of clocks. At the end of this time interval, the 82750PD completes any current bus cycle and then release the SFBC.
5	Quasi	This R/W bit determines whether the GRANT signal from the SFBC arbiter is synchronous or asynchronous. 0 = Synchronous 1 = Asynchronous
7-6	MMPR[1-0]	This 2-bit R/W field that defines the priority the 82750PD uses when requesting the SFBC. <u>MMPR[1-0] Priority</u> 0 0 High Priority 0 1 Medium Priority 1 0 Low Priority 1 1 No Priority (ILLEGAL! If No Priority is programmed and any memory cycle is attempted, the 82750PD HANGS.)

Table A-14. Core Interrupt Flag Register

Mnemonic: *cntflag*

Byte Offset: 100h

Access: R/O

Reset State: 01FFh

Bit(s)	15	14	13	12	11	10	9	8
Name	RSVD	VBI_F	MCINT_F	RSVD	MCINTO_F	OFI_F	EFI_F	Unused

Bit(s)	7—0
Name	Unused

Bit No.	Name	Description
8—0	RSVD	
9	EFI_F	Even Field Interrupt Flag.* This bit is set if an even field interrupt condition is detected.
10	OFI_F	Odd Field Interrupt Flag.*
11	MCINTO_F	Microcode Interrupt Overflow Flag. This bit is set if more than one microcode interrupt condition has occurred since the last time this register was read.
12	RSVD	Reserved
13	MCINT_F	Microcode Interrupt Flag.*
14	VBI_F	Vertical Blanking Interrupt Flag.*
15	RSVD	Reserved

\* This bit is set if its respective interrupt condition is detected, regardless of the state of its corresponding bit (corresponding by name, not by bit number), in the control register.

Table A-15. PCI POS2 Register

Mnemonic: *CMD0*

Address: 04h

Access: W/O

Reset State: 00h

Bit No.	7-2	1	0
Name	RSVD	MEMEN	IOEN

Bit No.	Name	Description
0	IOEN	Clearing this bit disables the host I/O registers.
1	MEMEN	Clearing this bit disables the "EMS" window, even if the EWE bit (bit 0 in the CFG 06) register is set.

Table A-16. Core Status Register

Name: cstatus

Byte Offset = 102:

Access: R/O

Reset State:

Bit(s)	15	14	13	12	11	10	9-8
Name	EFI_S	OFI_S	RSVD	RSDV	VBI_S	MCINT_S	Unusec

Bit(s)	7-4	3	2	1	0
Name	Unused	SYNC_S	PMON	FREEZE	HALT_S

Bit No.	Name	Description
0	HALT_S	Halt Status. This bit is set when the processor is halted because the HALT bit in the control register is set or because the HALT# pin is asserted
1	FREEZE	Freeze. This bit is set when the processor is waiting for the statistical decoder or one of the core FIFO's to become ready
2	PMON	Performance Monitor. This bit can be toggled by a special ALU opcode or by a special B source code. Set PMON to monitor the performance of microcode
3	SYNC_S	Synchronize Status. This bit is set if the internal synchronizers for the HREQ#/HALEN inputs are disabled
4-9		Unused
10	MCINT_S	Microcode Interrupt Status.*
11	VBI_S	Vertical Blanking Interrupt Status.*
12, 13	RSVD	Reserved
14	OFI_S	Odd Field Interrupt Status.*
15	EFI_S	Even Field Interrupt Status.*

\* The state of each these bits follows the state of its corresponding bit (corresponding by name, not by bit number) in the control register

Table A-17. "EMS" Configuration Registers

Mnemonic: EMSCFGn; n = 2-0

Address: CFG 08-06

Access: R/W

Reset State: 00h, 00h, 00h

Bit No.	7	6	5	4	3	2	1	0
EMSCFG2	EMS18	EMS17	EMS16	EMS15	EMS14	EMS13	EMS12	EMS11
EMSCFG1	EMS10	EMS9	EMS8	EMS7	EMS6	EMS5	EMS4	EMS3
EMSCFG0	EMS2	EMS1	EMS0	RSVD				EWE

Bit No.	Name	Description
7-0	EMS[18:0]	EMS[18:0] are the 19 MSBs of the base address of the 8-Kbyte "EMS" window. The MATCH field of the host address is compared to EMS[18:0] to detect an address that is to be mapped.
7-0		
7-5		
0	EWE	Setting this bit enables the "EMS" window, provided the MEM_ENABLE bit is set (see Section 5.2).

Table A-18. General Control Register

Mnemonic: GENCON

Address: REG6-BYTE2

Access: R/W

Reset State: 00h

Bit No.	7	6	5-0
Name	RSVD	DINT	RSVD

Bit No.	Name	Description
6	DINT	Disable Interrupt. Setting this bit disables the 82750PD interrupt. Clearing this bit enables the 82750PD to generate an interrupt.

Table A-19. General Status Register

Mnemonic: GENSTAT

Address: REG6-BYTE3

Access: R/O

Reset State: 00h

Bit No.	7-2	1	0
Name	RSVD	VINT	RSVD

Bit	Name	Description
6	VINT	VINT. This bit is set whenever the 82750PD interrupt signal is asserted

Table A-20. Input FIFO Control Register

Mnemonic: *inN-c*; N=1, 2

Address:

Access:

Reset State: Not available.

Bit No.	15-6	5	4	3	2	1	0
Name	RSVD	BY-32 MODE	CB	PF-OFF	AHOLD	INC/ DEC	WORD/ BYTE

Bit	Name	Description
0	WORD/BYTE	Word/Byte. This bit determines whether the Core input FIFO operates in word mode or byte mode 1 = Byte mode. The FIFO can start reading memory on any byte boundary. Each word read has data in the low byte and 00h in the high byte 0 = Word mode. The FIFO can start reading memory on any word boundary. Each word read has data in both the low high bytes
1	INC/DEC	Increment/Decrement. This bit determines the order in which bytes or words are read from the internal bus 1 = Decrement mode. The FIFO reads from the most significant to the least significant byte or word 0 = Increment mode. The FIFO reads from the least significant to the most significant byte or word



Table A-20. Input FIFO Control Register (Continued)

2	AHOLD	<b>Address Hold.</b> Setting this bit disables automatic incrementing/decrementing of $mN-lo$ and $mN-hi$ and prevents the Core Input FIFOs from double buffering the read data. At the end of an internal bus cycle the FIFO is updated with 64 bits of data. The Core Input FIFO will not issue another read request to the internal bus until there is a write to $mN-lo$ OR a roll-over/roll-under read access of the Core Input FIFO. If there is a write to $mN-lo$ the FIFO will then fetch data from the new location. If a roll-over/roll-under occurs, then a memory request will be issued to fetch data from the unchanged address.
3	PF-OFF	<b>Prefetch Off.</b> Setting this bit causes the Core Input FIFO to wait for a request to fetch a new quad word over the internal bus.  1 = PREFETCH ON mode. The Core Input FIFO prefetches successive quad words as necessary to keep its buffer full. Fetch addresses ascend or descend according to the INC/DEC bit.  0 = PREFETCH-OFF mode. The FIFO will still prefetch the first two quad words to fill its buffer (when started at a new address location) but will fetch a new quad word only when a read request is made to the FIFO for a value in the next unfetched quad word.
4	CB	<b>Circular Buffer.</b> Setting this bit enables the creation of a circular buffer in the SFB. The appropriate address bit on the internal bus (depending on the size of the circular buffer to be created) is cleared. The register pointers remain unchanged. The size of the circular buffer can be 64 Kbytes, 128 Kbytes, or 256 Kbytes, as determined by bits 2-0 of the Circular Buffer register <i>arcbuf</i> .
5	BY-32 MODE	<b>BY-32 MODE.</b> Setting this bit causes the pointer to increment or decrement by four bytes rather than two bytes.
15-9		Reserved bits

\* See A and B bus addresses

Table A-21. Indirect Address Register

Mnemonic: INDADDR

Address: REG4-BYTE3—REG4-BYTE0

Access: R/W

Reset State: 00000000h

Bit No.	31-0
Name	—

Bit No.	Name	Description
15-0	—	These four bytes are the 82750PD internal bus address for an indirect access of the 82750PD memory address space

Table A-22. Indirect Data Register

Mnemonic: INDDATA

Address: REG5-BYTE3—REG5-BYTE0

Access: R/W

Reset State: Uninitialized

Bit No.	31-0
Name	—

Bit No.	Name	Description
15-0		These four data bytes are data read from or written to the 82750PD memory for an indirect access of the 82750PD memory address space. The 82750PD address for the indirect access is in REG4-BYTE3—REG4-BYTE0

Table A-23. Message Configuration Register

Mnemonic: MCFG

Address: 00FF003C–00FF003F

Access: R/W

Reset State: See bit descriptions.

Bit No.	31–28	27–24	23	22–17	16
Name	V2CODE	V1CODE	PROTECT	0	EXTVBUS

Bit No.	15–12	11	10	9	8	7–6	5–4	3	2–0
Name	RSVD	CLK_EN	ARB_EN	DMSG_EN	MSG_EN	RSVD	CLKDIV	RSVD	MSG_ID

Bit No.	Name	Description																		
2–0	MSG_ID	Message ID. This field defines the ID of the 82750PD in the SynchroLink protocol.																		
3	0																			
5–4	CLKDIV	<p>Clock Division. When the 82750PD is the arbiter (ARB_EN = 1), these bits define the SynchroLink bus clock frequency (SLCLK) in terms of the SFBI clock frequency (MCLK) as listed in the following table.</p> <table> <tr> <th colspan="3">SynchroLink</th></tr> <tr> <th>Bit 5</th><th>Bit 4</th><th>Bus Freq. SLCLK</th></tr> <tr> <td>0</td><td>0</td><td>MCLK/2</td></tr> <tr> <td>0</td><td>1</td><td>MCLK/4</td></tr> <tr> <td>1</td><td>0</td><td>MCLK/6</td></tr> <tr> <td>1</td><td>1</td><td>MCLK/8</td></tr> </table>	SynchroLink			Bit 5	Bit 4	Bus Freq. SLCLK	0	0	MCLK/2	0	1	MCLK/4	1	0	MCLK/6	1	1	MCLK/8
SynchroLink																				
Bit 5	Bit 4	Bus Freq. SLCLK																		
0	0	MCLK/2																		
0	1	MCLK/4																		
1	0	MCLK/6																		
1	1	MCLK/8																		
7–6	0																			
8	MSG_EN	Message Enable. Setting this bit enables the 82750PD to send and receive messages. Clearing MSG_EN prevents the 82750PD from sending or receiving messages. The effect of changes in the MSG_EN bit do not take effect until a break (ten consecutive '0' bits) on the SynchroLink bus is detected.																		
9	DMSG_EN	Delay Message Enable. The value of this R/O bit follows the value written to the MSG_EN bit. DMSG_EN changes state only when the actual status of the 82750PD changes. (The actual change in status is delayed until the next break (ten consecutive '0' bits) on the two wire bus after a change in the MSG_EN bit.)																		
10	ARB_EN	Arbiter Enable. Setting this bit defines the 82750PD as the arbiter for the SynchroLink bus.																		
11	CLK_EN	Clock Enable. Setting this bit causes the 82750PD to drive the SynchroLink clock at a frequency specified by the CLKDIV bits (bits 5–4).																		

Table A-23. Message Configuration Register (Continued)

15-12	0	
16	EXTVBUS	<b>External VBUS.</b> Setting this bit enables the 82750PD VBUS to be driven from the external VBUSIN pins of the chip. Do not select this mode unless the host bus type is ISA or PCI. Furthermore, do not select this mode if you want to allow meta-interrupts to the 82750PD core. If the EXTVBUS is set, the VBUS codes generated by the SynchroLink interface are not driven to the 82750PD core. Instead, the states of the external VBUSIN pins are driven to the core via the VBUS.
22-17	0	
23	PROTECT	<b>PROTECT.</b> Setting/clearing this bit disables/enables further writes to this register. During normal operation, this bit can be cleared only by writing bits 31-24 with data that matches the data already in bits 31-24.

Table A-24. Micro Channel POS3 Register

Mnemonic: —

Address: xx02h\*

Access: W/O

Reset State: 00h

Bit No.	7-1	0
Name	RSVD	CDEN

Bit No.	Name	Description
0	CDEN	CDEN = IO_ENABLE = MEM_ENABLE. Clearing this bit disables the host I/O registers, and disables the "EMS" window, even if the EWE bit (bit 0 in the CFG 06 register) is set.

\* xx denotes eight don't-care bits

Table A-25. Micro Channel POS3 Register

Mnemonic: —

Address: xx03h\*

Access: W/O

Reset State: 75h

Bit No.	7-0
Name	IOB[7:0]

Bit No.	Name	Description
7-0	IOB[7:0]	IOB[7:0] = IO_BASE[7:0]**

\* xx denotes eight don't-care bits

\*\* IO Base[8] is latched from straps or jumpers at reset.

Table A-26. Match Service Register

Mnemonic:

Address: 00FF0038h-00FF003B

Access: R/W

Reset State: See bit descriptions.

Bit No.	31-25	27-18	17-16	15-12	11-2	1-0
Name	0	SRV_MASK	0	0	SRV_DATA	0

Bit No.	Name	Description
31-25	0	The 0 denotes bits that should always be written with 0. The actual value written is ignored. These bits will be 0 when read.
27-18	SRV_MASK	SRV_MASK is a 10-bit R/W field that is used to select which bits of the service number are compared. If a bit in SRV_MASK = 1, then the corresponding bit in the service number of the service request will be compared with the corresponding bit in the SRV_DATA field. If a bit in SRV_MASK = 0 then the corresponding bit in the service number of the service request will be ignored. These bits are uninitialized by RESET.
17-16	0	The 0 denotes bits that should always be written with 0. The actual value written is ignored. These bits will be 0 when read.
15-12	0	The 0 denotes bits that should always be written with 0. The actual value written is ignored. These bits will be 0 when read.
11-2	SRV_DATA	SRV_DATA is a 10-bit R/W field that is compared with the service number of service request messages to determine if the message is an incoming message. These bits are uninitialized by RESET.
1-0	0	The 0 denotes bits that should always be written with 0. The actual value written is ignored. These bits will be 0 when read.

Table A-27. Message Status Register

Mnemonic: MSTATUS

Address: 00FF0030h-00FF0033

Access: R/O

Reset State: 00h

Bit No.	31-25	24	23-22	21-20	19-18	17-16
Name	0	RSRV STAT	XCVRB STAT	XCVRA STAT	XCVR9 STAT	XCVR8 STAT

Bit No.	15-14	13-12	11-10	9-8	7-6	5-4	3-2	1-0
Name	XCVR7 STAT	XCVR6 STAT	XCVR5 STAT	XCVR4 STAT	XCVR3 STAT	XCVR2 STAT	XCVR1 STAT	XCVR0 STAT

Bit	Name	Description
$2n+1-2n$	XCVR $n$ STAT	Transceiver $n$ Status ( $n = 0, 1, 2, \dots, 9, A, B$ ). Each 2-bit field reflects the status of bits in the corresponding XCVR $n$ register. The specific bits whose status is shown depends on the value of the MSG bit in the same XCVR $n$ register. The selected XCVR $n$ bits take their values from the XCVR $n$ bits as follows:  <div style="display: flex; justify-content: space-between;"> <span>MSG</span> <span>XCVR<math>n</math> STAT[2<math>n</math>+1]</span> <span>XCVR<math>n</math> STAT[2<math>n</math>]</span> </div> <div style="display: flex; justify-content: space-between;"> <span>0</span> <span>FMATCH</span> <span>TXIPR</span> </div> <div style="display: flex; justify-content: space-between;"> <span>1</span> <span>NACK OR* COMPLETE</span> <span>TXIPR</span> </div> <p>* Logical inclusive OR</p>
24	RSRV STAT	Receive Service Status. This bit reflects the state of the VALID bit in the RSRV register.
31-25	—	These bits are clear.

Table A-28. Core Output FIFO Control Register

Mnemonic: outN-c; N=1,2

Address: Core Register

Access: R/W

Reset State: Not available.

Bit No.	15-6	5	4	3	2	1	0
Name	RSVD	BY-32 MODE	FORCE-LSB ENABLE	FORCE LSB VALUE	AHOLD	DEC INC#	BYTE WORD#

Bit No	Name	Description
0	BYTE/ WORD#	<b>Byte/Word#.</b> This bit determines whether the Core Output FiFO operates in word mode or byte mode. 1 = Byte mode The FIFO can start writing memory on any byte boundary 0 = Word mode The FIFO can start writing memory on any word boundary
1	DEC/INC#	<b>Decrement/Increment#.</b> This bit determines the order in which bytes or words are written to the 82750PD internal bus 1 = Decrement mode. The FIFO writes from the most significant to the least significant byte or word. 0 = Increment mode. The FIFO writes from the least significant to the most significant byte or word
2	AHOLD	<b>Address Hold.</b> Setting this bit disables automatic incrementing/decrementing of outN-h and outN-lo. The FIFO continues to write to a single quad word in the 82750PD address space.
3	FORCE LSB VALUE	<b>Force LSB Value.</b> The value of this bit is the value of the least significant bit written to each byte, provided that the FORCE LSB ENABLE bit is set
4	FORCE LSB ENABLE	<b>Force LSB Enable.</b> Setting this bit forces the least significant bit of each byte written to the 82750PD internal bus to be a '1' or a '0', as specified by the FORCE LSB VALUE bit
5	BY-32 MODE	<b>By-32 Mode</b> Setting this bit causes the pointer to increment or decrement by four bytes rather than 2 bytes
15-6	RSVD	

\* See A and B bus addresses

Table A-29. "EMS" Page Address Registers.

Mnemonic: PAR $n$ ;  $n = 3-0$ Address: REG1-BYTES3-2, REG1-BYTES1-0,  
REG0-BYTES3-2, REG0-BYTES1-0

Access: See bit descriptions.

Reset State: 0000h

Bit No.	15-3	2	1	0
Name	PAR	0	0	0

Bit No.	Name	Description
3-0	0	These bits are wired to '0'. They read as '0' and should be written as '0'.
15-3	PAR	These R/W bits are the 13 MSBs of the base address of a 2 Kbyte page in the 82750PD memory address space. A host access to page $n$ of the "EMS" window is redirected to this page (provided the "EMS" mode is enabled).



Table A-30. Pixel Interpolator Control Register

Mnemonic: *pixint-c*

Address\*

Access: R/W

Reset State: Not available

Bit No.	15	14	13	12	11	10	9:8
Name	RSVD	PIPELINING	PHASE	RSVD	PAIRING	RESET	MODE SELECT

Bit No.	7:4	3:0
Name	VERTICAL WEIGHT	HORIZ WEIGHT

Bit No.	Name	Description
3-0	HORIZ WEIGHT	<b>Horizontal Weighting, h</b> These bits contain the horizontal weighting, expressed as the numerator of a fraction which is an even multiple of 1/16
7-4	VERTICAL WEIGHT	<b>Vertical Weighting, v</b> These bits contain the vertical weighting, expressed as the numerator of a fraction which is an even multiple of 1/16
9-8	MODE SELECT	<b>Pixel Interpolator Operating Modes</b> 0 0 = Random-2D (nonsequential) Use this mode if the pixels are not in horizontal rows or if the h and v weightings are not constant over all pixels 0 1 = Sequential-2D: the h and v weightings are constant for all interpolations 1 0 = RSVD 1 1 = RSVD
10	RESET	Setting this bit resets the pixel interpolator
11	PAIRING	<b>Pixel Pairing</b> 0 = The pixel interpolator outputs individual 8-bit pixels 1 = The interpolator outputs 16-bit pixel pairs comprising adjacent pixels, provided that the MODE SELECT bits select Sequential 2-D mode
12	RSVD	

Table A-30. Pixel Interpolator Control Register (Continued)

13	PHASE	<b>Bitmap Phase</b> 0 = In phase The first two output pixel pairs are grouped into one 16-bit pair with the first pixel in the least significant byte 1 = Out of phase The first pixel is placed in the most significant byte of the first pixel pair; the least significant byte of the first pixel pair contains invalid data															
14	PIPELINING	<b>Pipelining</b> Setting this bit reduces the pipelining delay which also depends on the PHASE bit <div style="text-align: center;">Pipeline Delay in</div> <table> <tr> <th>Bit 14</th><th>Bit 13</th><th>Output Pixels</th></tr> <tr> <td>0</td><td>0</td><td>6</td></tr> <tr> <td>0</td><td>1</td><td>7</td></tr> <tr> <td>1</td><td>0</td><td>2</td></tr> <tr> <td>1</td><td>1</td><td>3</td></tr> </table>	Bit 14	Bit 13	Output Pixels	0	0	6	0	1	7	1	0	2	1	1	3
Bit 14	Bit 13	Output Pixels															
0	0	6															
0	1	7															
1	0	2															
1	1	3															
15	RSVD																

\* See A bus and B bus addresses

Table A-31. Read FIFO Address Counter Register

Mnemonic: RFCNTR

Address: REG3-BYTE3 with BS = 2-0

Access: R/W

Reset State: 00h

Bit No.	23-16	15-8	7-0
Name	CNTRH (BS = 2)	CNTRM (BS = 1)	CNTRL (BS = 0)

Bit No	Name	Description
7-0	CNTRL	Low byte of Read FIFO Address Counter
15-8	CNTRM	Middle byte of Read FIFO Address Counter
23-16	CNTRH	High byte of Read FIFO Address Counter

Table A-32. Read FIFO Control Register

Mnemonic: RFCON

Address: REG3-BYTE2

Access: R/W

Reset State: 00h

Bit No.	7	6	5	4	3	2	1	0
Name	AUTO	TEST	R FULL* W RSVD	FAST	R EMPTY* W RSVD	BS2	BS*	BS0

\* R = Read. W = Write

Bit No.	Name	Description																														
0-2	BS0-BS2	<p><b>Byte Select</b> These bits specify the register byte that is to be accessed through the Selected Byte Register (REG2-BYTE3). BS2 is a "don't care"; it is not decoded by the hardware</p> <table><tr><th>BS2</th><th>BS1</th><th>BS0</th><th>BS</th><th>WRITE</th><th>READ</th></tr><tr><td>X</td><td>0</td><td>0</td><td>0</td><td>CNTRL</td><td>CNTRL</td></tr><tr><td>X</td><td>0</td><td>1</td><td>1</td><td>CNTRM</td><td>CNTRM</td></tr><tr><td>X</td><td>1</td><td>0</td><td>2</td><td>CNTRH</td><td>CNTRH</td></tr><tr><td>X</td><td>1</td><td>1</td><td>3</td><td>NOP</td><td>RFTEST</td></tr></table> <p>Writing these bits selects a byte in the "WRITE" column: a byte (CNTRx) of the 24-bit SFB address. Reading these bits selects a listing in the "READ" column: a byte (CNTRx) of the 24-bit SFB address, a byte (DHx) from the data holding register, or the WFTST register</p> <p>For a read of the RFTEST register see the RFTEST register table</p>	BS2	BS1	BS0	BS	WRITE	READ	X	0	0	0	CNTRL	CNTRL	X	0	1	1	CNTRM	CNTRM	X	1	0	2	CNTRH	CNTRH	X	1	1	3	NOP	RFTEST
BS2	BS1	BS0	BS	WRITE	READ																											
X	0	0	0	CNTRL	CNTRL																											
X	0	1	1	CNTRM	CNTRM																											
X	1	0	2	CNTRH	CNTRH																											
X	1	1	3	NOP	RFTEST																											
3	EMPTY	<p><b>Empty.</b> The EMPTY bit reflects the status of the data holding registers, R1 and R2. The EMPTY bit is set if neither R1 nor R2 contains data. In AUTO mode, the EMPTY bit would normally be set only for a short time after the SFB address has been loaded. The EMPTY bit and the FULL bit are not complements. If the host reads the RFDATA register when the EMPTY bit is set, the access becomes a "slow access," i.e., host wait states are inserted until the read is finished. See also the FAST bit.</p>																														
4	FAST	<p><b>Fast</b> If the fast bit is set and the host reads the RFDATA register, the access becomes a "fast access" provided these additional conditions are met.</p> <p>The FIFO is in AUTO mode. The FIFO is empty (EMPTY = 1) The RECNR register is pointing to an even dword</p>																														

Table A-32. Read FIFO Control Register (Continued)

5	FULL (Read)  TCLK (Write)	<p>Full (Read) This bit is set if register R1 has data and the high byte (DH3) of register R2 has data, i.e., has not been read. Software should verify that the FIFO is "full" before it alters the RFCNTR or RCON register.</p> <p>If the FULL bit is clear, the Read FIFO contains some residual data. When in the AUTO mode, the Read FIFO always contains some residual data after the host reads the RFDATA register. To purge residual data, take the FIFO out of AUTO mode and read (at least) byte DH3 of R2.</p> <p>TCLK. TCLK must be clear for proper FIFO operation in its normal mode. Set TCLK only for diagnostics. If TCLK is set, the FIFO address in RFCNTR is not incremented in AUTO mode. If TCLK changes from set to clear, the RFCNTR register is incremented by 4.</p>
6	Test	<p>Test. The bit is used for diagnostics. Setting this bit initializes the FIFO and prevents the internal bus acquisition logic from requesting the bus. This allows the host to access the Read FIFO registers without triggering a read cycle. Reading an empty Read FIFO with TEST = 1 returns erroneous data. Furthermore, IOCHRDY is never asserted (the read cycle is never initiated).</p>
7	AUTO	<p>Automatic Increment. When the AUTO bit is set, the RFCNTR register increments by four after each read cycle of the internal bus, thus setting up the address counter for reading the next dword. When the Auto bit is clear, the RFCNTR register does not increment automatically. To read a single dword, word, or byte, the AUTO bit should be clear. The Read FIFO accesses the same SFB address repeatedly until a new address is written to the RFCNTR register.</p>

Table A-33. Read FIFO Data Register

Mnemonic: RFDATA

Address: REG3-BYTE1—REG3-BYTE0

Access: R/W

Reset State: Uninitialized

Bit No.	15—0
Name	—

Bit	Name	Description
15—0	—	Data retrieved from SFB is read from this register.

Table A-34. Read FIFO Selected Byte Register

Mnemonic: RFSELB

Address: REG3-BYTE3

Access: R/W

Reset State: See description below

This register is used to access the registers specified by the BS2-BS0 bits of the RFCON register. See the RFCON register. The registers accessed via RFSELB have the following reset values

CNTRL	CNTRM	CNTRH	00h
	RFTEST		43hh

Table A-35. Read FIFO Test Register

Mnemonic: RFTEST

Address: REG3-BYTE3, BS = 3 or 7

Access: R/O

Reset State: 43h

Bit No.	7	6	5	4	3	2	1	0
Name	REQ	SEL	RFRES	FAST	CNT2	NXTFAST	FFB	FFA

Bit No.	Name	Description
0	FFA	Flipflop A. If this bit is set, data holding register R2 (host side) contains valid data.
1	FFB	Flipflop B. If this bit is set, data holding register R1 (SFB side) contains valid data.
2	NXTFAST	Next Fast. If this bit is set, the Read FIFO is requesting a 64-bit internal bus cycle.
3	CNT2	CNT2. This bit is bit 2 of the RFCNTR register.
4	FAST	FAST. This is the FAST bit (bit 4) of the RFCON register.
5	RFRES	Read FIFO Reset. If this bit is set, the Read FIFO is being reset. The FIFO enters its reset state upon reset of the 82750PD or when the host writes the low byte of the RFCNTR register. The FIFO exits its reset state when the host writes the high byte of the RFCNTR register.
6	SEL	Select. If this bit is clear, the internal UHBI arbiter has selected the Read FIFO.
7	REQ	Request. If this bit is set, the Read FIFO is requesting use of the 82750PD internal bus.

Table A-36. Receive Service Register

Mnemonic: RSRV

Address: 00FF0034-00FF0037

Access: See bit descriptions.

Reset State: See bit descriptions

Bit No.	31-25	24	23-17	16	15-12	11-2	1-0
Name	RSVD	IINT	RSVD	VALID	FCODE	SERV_NUM	RSVD

Bit	Name	Description
1-0	RSVD	-
11-2	SERV_NUM	<b>Service Number</b> (R/O; uninitialized by reset) This field latches the service number (packet bits 10-13 and 15-20) from the received message
15-12	FCODE	<b>Function Code</b> (R/O; uninitialized by reset) This 4-bit field always contains 0Eh (the service request code) after a service request is received. These bits are uninitialized by reset
16	VALID	<b>Valid</b> . (R/W; reset state = 0). This can be used by 82750PD core microcode or the host to determine if a completion message was received. The SynchroLink hardware sets this bit when a service completion message is received and acknowledged. An incoming completion message is defined as a message satisfying these criteria: <ul style="list-style-type: none"> <li>The function code (bits 6-9) = 0Eh (service request)</li> <li>The service number (packet bits (10-13, 15-20)) is a <i>recognized service number</i> (see the Match Service register (MSRV) described in Section 6.4.8)</li> </ul> <p>If the message does not meet these criteria, the VALID bit is not set and the service acknowledge bit (bit 22) in the message packet is not set.</p> <p>The 82750PD core microcode and the host can reset the valid bit, but writing a '1' has no effect. If software clears this bit and hardware sets this bit in the same clock, the bit is cleared</p>
23-17	RSVD	
24	IINT	<b>Incoming Message Interrupt</b> (R/W; reset state = 0) Setting this bit enables a NOT-VALID to VALID transition (bit 6) to generate the V1CODE VBUS code
31-25	RSVD	

Table A-37. Statistical Decoder Control Register

Mnemonic: *stat-c*

Address\*

Access: R/W

Reset State: Not available.

Bit No.	15	14	13	12:8
Name	POL	RSVD*	CB	SVAL

Bit No.	7	6	5	4	3	2:0
Name	SHORT	END	TEST	WRITE	RSVD*	STNDX

Bit No.	Name	Description
2:0	STNDX	<b>Starting Index.</b> These bits specify the starting index value of the code description table. The table access begins at this value.
3	RSVD	
4	WRITE	<b>Write.</b> The WRITE bit and the TEST bit control reads and writes to the code description table as follows: <b>TEST WRITE</b> 0 0 After writing to the code description table and clearing these bits, the new values in the c. d. table will take effect. 0 1 Enables a write to the c. d. table 1 0 Enables a read of the c. d. table 1 1 Reserved
5	TEST	<b>Test.</b> See description for WRITE.
6	END	<b>End.</b> Setting this bit invokes the END bit mode. The '1' trailing the run-in sequence is dropped from the code.
7	SHORT	<b>Short.</b> Setting this bit invokes the SHORT mode. The values of $2^X$ are the same for every entry in the code description table.
8-12	SVAL	<b>Short Value.</b> If the SHORT bit is set, these bits specify the value of $2^X$ that is used in the code description table in SHORT mode.
13	CB	<b>Circular Buffer.</b> Setting this bit enables the creation of circular buffers of size 64 Kbytes, 128 Kbytes, or 256 Kbytes.
14	RSVD	
15	POL	<b>Polarity.</b> This bit specifies the polarity of the bits in the run-in sequence.

\* See A bus and B bus addresses

Table A-38. Write FIFO Address Counter Register

Mnemonic: WFCNTR

Address: REG2-BYTE3 with  
BS = 2-0 (R/W) or 6-4 (W)

Access: R/W

Reset State: 0000h

Bit No.	23-16	15-8	7-0
Name	CNTRH (BS = 2 (W/R), BS = 6 (W))	CNTRM (BS = 1 (W/R), BS = 5 (W))	CNTRL (BS = 0 (W/R), BS = 4 (W))

Bit No.	Name	Description
7-0	CNTRL	Low byte of Write FIFO Address Counter
15-8	CNTRM	Middle byte of Write FIFO Address Counter
23-16	CNTRH	High byte of Write FIFO Address Counter



Table A-39. Write FIFO Control Register

Mnemonic: WFCON

Address: REG2-BYTE2

Access: R/W

Reset State: 08h

Bit No.	7	6	5	4	3	2	1	0
Name	AUTO	TEST	R FULL* W TCLK	FAST	R EMPTY* W RSVD	RS2	RS1	RS0

\* R = Read, W = Write

Bit No.	Name	Description																																																						
0-2	BS0-BS2	<p><b>Byte Select.</b> These bits specify the register byte that is to be accessed through the Selected Byte Register (REG2-BYTE3)</p> <table border="1"> <thead> <tr> <th>BS2</th><th>BS1</th><th>BS0</th><th>BS</th><th>WRITE</th><th>READ</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>CNTRL</td><td>CNTRL</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>CNTRM</td><td>CNTRM</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>2</td><td>CNTRH</td><td>CNTRH</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>3</td><td>FLUSH</td><td>WFTST</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>4</td><td>CNTRL</td><td>DH0</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>5</td><td>CNTRM</td><td>DH1</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>6</td><td>CNTRH</td><td>DH2</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>7</td><td>FLUSH</td><td>DH3</td></tr> </tbody> </table> <p>Writing these bits selects a byte in the "WRITE" column a byte (CNTRx) of the 24-bit SFB address or the FLUSH Register. Reading these bits selects a listing in the "READ" column a byte (CNTRx) of the 24-bit SFB address, a byte (DHx) from the data holding register, or the TEST register.</p> <p>Writing any value to the FLUSH register flushes the data holding register (R1, R2). For a read of the WFTST register see the WFTST register table.</p>	BS2	BS1	BS0	BS	WRITE	READ	0	0	0	0	CNTRL	CNTRL	0	0	1	1	CNTRM	CNTRM	0	1	0	2	CNTRH	CNTRH	0	1	1	3	FLUSH	WFTST	1	0	0	4	CNTRL	DH0	1	0	1	5	CNTRM	DH1	1	1	0	6	CNTRH	DH2	1	1	1	7	FLUSH	DH3
BS2	BS1	BS0	BS	WRITE	READ																																																			
0	0	0	0	CNTRL	CNTRL																																																			
0	0	1	1	CNTRM	CNTRM																																																			
0	1	0	2	CNTRH	CNTRH																																																			
0	1	1	3	FLUSH	WFTST																																																			
1	0	0	4	CNTRL	DH0																																																			
1	0	1	5	CNTRM	DH1																																																			
1	1	0	6	CNTRH	DH2																																																			
1	1	1	7	FLUSH	DH3																																																			
3	R EMPTY W RSVD	<p><b>EMPTY.</b> This bit is set, when either of two conditions hold:</p> <ul style="list-style-type: none"> <li>(i) the host has not written to the data holding register since the FIFO has written holding registers R2 and R1 to the SFB, OR</li> <li>(ii) The FIFO has just been flushed</li> </ul>																																																						
4	FAST	<p><b>FAST.</b> If FAST = 1 and the FIFO is full, then attempting to write data to an even dword address causes the 82750PD to execute a fast internal bus cycle (i.e., the data in both R1 and R2 are written to the SFB). If FAST = 0, the 82750PD executes only normal bus cycles.</p>																																																						

Table A-39. Write FIFO Control Register (Continued)

5	R FULL/ W TCLK	<b>FULL TCLK. READ</b> The FULL bit is set if (i) data holding register R2 holds valid data which has not yet been written to the SFB AND (ii) the host has written data to (at least) the most significant byte of R1 <b>WRITE.</b> Setting this bit prevents incrementation (by 4) of the WFCNTR register when the AUTO bit is set. If TCLK changes from set to clear the address counter is incremented by 4. TCLK must be clear for operation in normal mode; set this bit only in TEST mode.
6	TEST	<b>TEST.</b> Setting this bit invokes TEST mode, which is for diagnostics. In TEST mode, the FIFO is prevented from requesting the internal bus; the host can access the Write FIFO registers without triggering a write cycle. Clear this bit for normal operation.
7	AUTO	<b>Automatic.</b> Setting this bit causes the FIFO WFCNTR register to increment by 4 after each internal bus write data cycle. If AUTO is clear, the FIFO will access the same location repeatedly, until a new address is written to the address counter.

Table A-40. Write FIFO Data Register

Mnemonic: WFDATA

Address: REG2-BYTE1—REG2-BYTE0

Access: R/W

Reset State: Uninitialized

Bit No.	15—0
Name	—

Bit No.	Name	Description
15—0		Data to be written to the SFB is written to this register.

Table A-41. Write FIFO Selected Byte Register

Mnemonic: WFSELB

Address: REG2-BYTE3

Access: R/W

Reset State: 00h

This register is indirectly addressed to access the registers specified by the BS2-BS0 bits of the Write FIFO Control register (WFCN). See the WFCN register. The registers accessed via WFSELB have the following reset values:

CNTRL, CNTRM, CNTRH 00h

DH3—DH0 uninitialized

WFTST 0FFh

Table A-42. Write FIFO Test Register

Mnemonic: WFTST

Address: REG2-BYTE3 with BS = 3

Access: R/O

Reset State: 0FFh

Bit No.	7-4	3-0
Name	R1DH[7:4]	R2DH[3:0]

Bit No.	Name	Description
3-0	R2DH $n$	The R2DH $n$ bit ( $n = 0, 1, 2, 3$ ) of data holding register R2 is set if bit $n$ of data holding register R2 is empty.
7-4	R1DH $n$	The R1DH $n$ bit ( $n = 0, 1, 2, 3$ ) of data holding register R1 is set if bit $n$ of data holding register R2 is empty.

Table A-43. Transceiver Registers

Mnemonic: XCVRn; n = 00h-0Bh

Address: 00FF0000h-00FF002Fh

Access: R/W

Reset State: See bit descriptions.

Bit No.	31	30	29	28	27	26	25	24
Name	MSG	RXEN	BINT2	RSVD	CINT	BINT	NINT	TXIN

Bit No.	23-21	20	19	18	17	16
Name	RSVD	SUCCESS	COMPLETE	FMATCH	NACK	TXIPR

Bit No.	15-12	11-0
Name	FCODE	DATA

Bit No.	Name	Description
11-0	DATA	DATA (Uninitialized at reset). These bits correspond to the data bits in the data packet (packet bits 10-13, 15-22). When the register is in the Broadcast Transmit mode (RXEN = 0), software can write these bits. When the register is in the Broadcast Receive mode (RXEN = 1), hardware can write these bits.
15-12	FCODE	Function code (Uninitialized at reset). This field contains the function code.
16	TXIPR	TXIPR (reset state = 0). Software sets this bit to initiate a transmission (write the FCODE and DATA fields onto the SynchroLink bus). Hardware clears this bit when the transmission is complete regardless of whether the transmission is acknowledged.
17	NACK	Not Acknowledged (reset state = 0). This bit is used only in a service-request/service-completion operation (MSG = 1). Hardware sets this bit in the source's register to indicate that the target does not acknowledge the service request.
18	FMATCH	Function Code Match (reset state = 0). This bit is used by a device that is in broadcast receive mode (RXEN = 1). Software clears this bit when it begins listening for a broadcast message with a specific function code. Hardware sets this bit to indicate that a message with that function code has been received.
19	COMPLETE	Complete (reset state = 0). This bit is used only in a service-request/service-completion transaction (MSG = 1). Software must clear this bit in the source's register when it transmits a service request (sets TXIPR). Hardware sets this bit when a corresponding completion message is received, i.e. when: (i) The FCODE in the service completion message is 0Fh, and (ii) the service number (packet bits 10-13, 15-20) in the completion message matches the service number in the transceiver register that was used to send the service request.

Table A-43. Transceiver Registers (Continued)

20	SUCCESS	<b>Success</b> (reset state = 0). This bit is set by hardware to indicate that a service has been successfully completed. (When a service completion message causes the COMPLETE bit to be set, the value of the SUCCESS bit is latched from packet bit 22 of the message.)															
21-23	RSVD																
24	TXINT	<b>Transmit Interrupt Enable</b> (reset state = 0). If this bit is set when hardware clears the TXIPR bit, a meta interrupt and a V1CODE signal are generated.															
25	NINT	<b>Nack Interrupt Enable</b> (reset state = 0). If this bit is set when hardware sets the NACK bit, a meta interrupt and a V1CODE signal are generated.															
26	BINT	<b>B Interrupt Enable</b> (reset state = 0). If this bit is set when hardware clears the FMATCH bit, a meta interrupt and a V1CODE signal are generated.															
27	CINT	<b>Complete Interrupt Enable</b> (reset state = 0). If this bit is set when hardware clears the COMPLETE bit, a meta interrupt and a V1CODE signal are generated.															
28	RSVD																
29	BINT2	<b>B2 Interrupt Enable</b> (reset state = 0). If this bit is set when hardware sets the FMATCH bit, a meta interrupt and a V2CODE signal are generated.															
30	SMSG	<b>Service Message</b> (reset state = 0). Setting this bit puts this register into service request and completion mode. Clearing this bit put this register into broadcast send and receive mode. This bit and the RXEN bit put this register into the following modes: <table border="1"> <thead> <tr> <th>SMSG</th> <th>RXEN</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Broadcast Transmit</td> </tr> <tr> <td>0</td> <td>1</td> <td>Broadcast Receive</td> </tr> <tr> <td>1</td> <td>0</td> <td>Service Request (send)</td> </tr> <tr> <td>1</td> <td>1</td> <td>Service Completion (receive)</td> </tr> </tbody> </table>	SMSG	RXEN		0	0	Broadcast Transmit	0	1	Broadcast Receive	1	0	Service Request (send)	1	1	Service Completion (receive)
SMSG	RXEN																
0	0	Broadcast Transmit															
0	1	Broadcast Receive															
1	0	Service Request (send)															
1	1	Service Completion (receive)															
31	RXEN	<b>Transmit Enable</b> (reset state = 0). Setting this bit enables the SynchroLink hardware (only) to write to the data field. Clearing this bit enables the 82750PD software (only) to write to the data field. See also the SMSG bit.															

## Appendix B

### Logical and Device Addresses

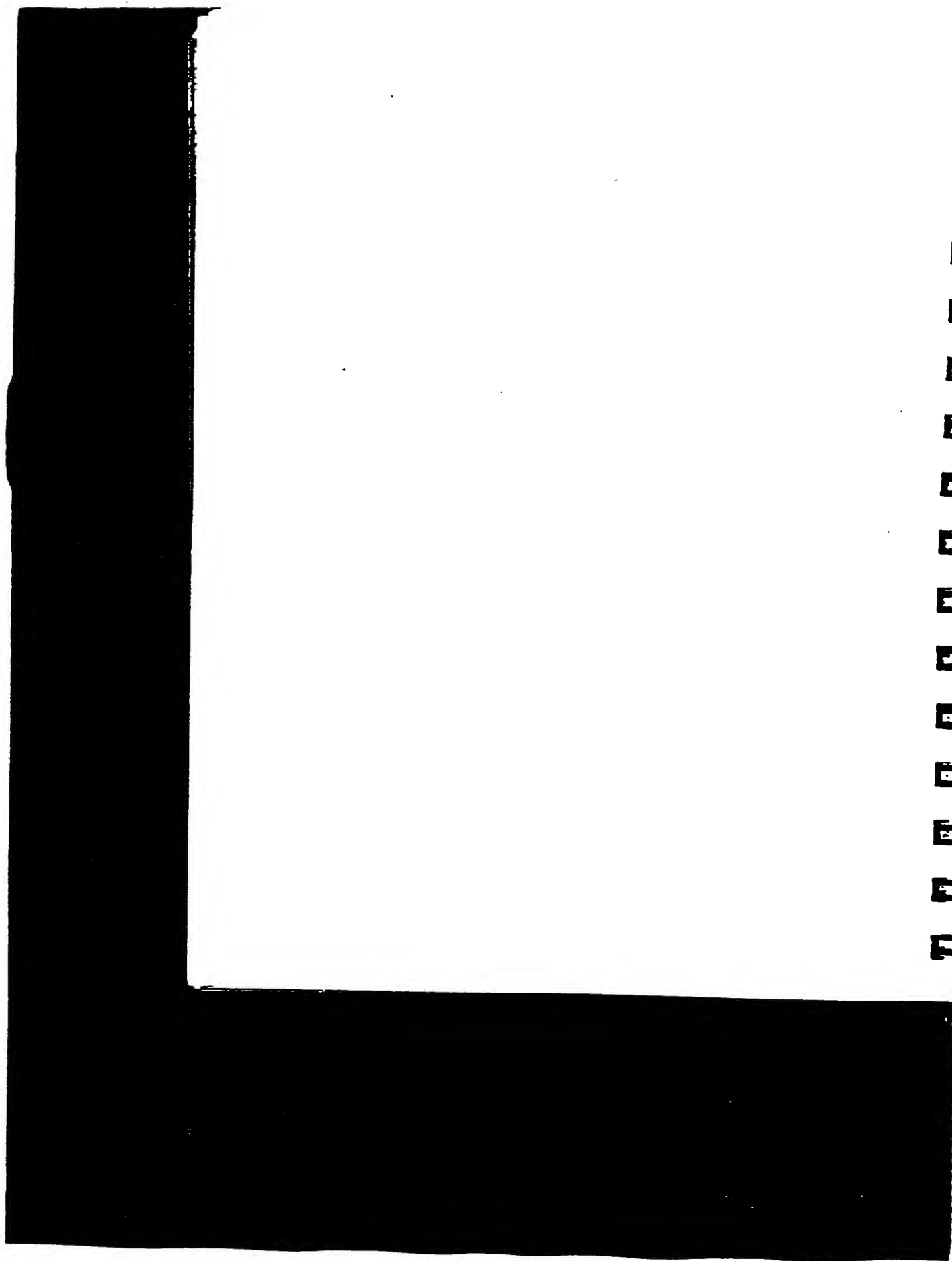
The 82750PD address bits are used to address memory on the SFB1. The exact association of 82750PD address bits to DRAM/VRAM bits depends on the exact DRAM/VRAMs used. Table B-1 details this association.

Table B-1. Association of 82750PD Address Bits to DRAM/VRAM Bits

MEMSIZE (Mbytes)	MEMTYPE*	Interleave A24=1(yes)	Bank Select (CAS[3:0])	Row Address (MA[9:0])	Column Address (MA[9:0])	Byte Select (WE[7:0])	Bus Width
010 (1-4)	0	A24=0	A[21:20]	A[20:11]	A[11:2]	A[1:0]	32-bit
010 (1-4)	0	A24=1	A[20:19]	A[19:10]	A[10:2, 22]	A[1:0]	32-bit
111(2-8)	0	A24=0	A[22:21]	A[21:12]	A[12:3]	A[2:0]	64-bit
111(2-8)	0	A24=1	A[21:20]	A[20:11]	A[11, 3, 22]	A[2:0]	64-bit
010 (1-4)	1	A24=0	A[21:20]	A[10, 19, 11]	A[11:2]	A[1:0]	32-bit
010 (1-4)	1	A24=1	A[20:19]	A[9, 18, 10]	A[10, 2, 22]	A[1:0]	32-bit
111(2-8)	1	A24=0	A[22:21]	A[11, 20, 12]	A[12:3]	A[2:0]	64-bit
111 (8MB)	1	A24=1	A[21:20]	A[10, 19, 11]	A[11, 3, 22]	A[2:0]	64-bit

MEMTYPE: 0 = symmetrical, 1 = non-symmetrical

The 82750PD accommodates both symmetrical and non-symmetrical DRAMs/VRAMs, which are denoted by MEMTYPE (0 = symmetrical, 1 = non-symmetrical). Symmetrical memories use a 9-row, 9-column address scheme, while non-symmetrical memories use a 10-row, 8-column scheme. For the non-symmetrical memory, the chip must have ten address pins (MA[9:0]). For symmetrical memories, only MA[8:0] are required to generate the RAM address. For non-symmetrical memories, MA[9:0] and MA[7:0] are used. For MEMTYPE = 1, the board can use symmetrical as well as a non-symmetrical RAM by mapping the unused column MA[8] to the row MA[9].



## Association of Logical and Physical Addresses

Table B-2 shows an example of how 82750PD addresses map to physical DRAM/VRAM bytes. This example uses MEMTYPE = 0 and covers MEMSIZE = 2MB, 4MB, 6MB, 8MB. Note that accesses to addresses, for which physical memory does not exist (e.g., above 2MB for MEMSIZE = 2MB), result in aliasing to existing memory or non-existing banks. In either case, the memory cycle completes and the 82750PD does not hang.

B-2

ATI018011



Table B-2. 82750PD Internal Bus Address to DRAM/VRAM Physical Address Mapping

	Logical 82750PD Address	Memory bank	ROW address	Column address
Unmapped Addresses	0000 0000 h	1st bank 256Kx64	0 0000 0000 b	0 0000 0000 b
	0000 0008 h	1st bank 256Kx64	0 0000 0000 b	0 0000 0001 b
	0000 1000 h	1st bank 256Kx64	0 0000 0001 b	0 0000 0000 b
	0010 0000 h	1st bank 256Kx64	1 0000 0000 b	0 0000 0000 b
	001F FFFB h	1st bank 256Kx64	1 1111 1111 b	1 1111 1111 b
	0020 0000 h	2nd bank 256Kx64	0 0000 0000 b	0 0000 0000 b
	0020 0008 h	2nd bank 256Kx64	0 0000 0000 b	0 0000 0001 b
	0020 1000 h	2nd bank 256Kx64	0 0000 0001 b	0 0000 0000 b
	0030 0000 h	2nd bank 256Kx64	1 0000 0000 b	0 0000 0000 b
	003F FFFB h	2nd bank 256Kx64	1 1111 1111 b	1 1111 1111 b
	0040 0000 h	3rd bank 256Kx64	0 0000 0000 b	0 0000 0000 b
	0040 0008 h	3rd bank 256Kx64	0 0000 0000 b	0 0000 0001 b
	0040 1000 h	3rd bank 256Kx64	0 0000 0001 b	0 0000 0000 b
	0050 0000 h	3rd bank 256Kx64	1 0000 0000 b	0 0000 0000 b
	005F FFFB h	3rd bank 256Kx64	1 1111 1111 b	1 1111 1111 b
	0060 0000 h	4th bank 256Kx64	0 0000 0000 b	0 0000 0000 b
	0060 0008 h	4th bank 256Kx64	0 0000 0000 b	0 0000 0001 b
	0060 1000 h	4th bank 256Kx64	0 0000 0001 b	0 0000 0000 b
	0070 0000 h	4th bank 256Kx64	1 0000 0000 b	0 0000 0000 b
	007F FFFB h	4th bank 256Kx64	1 1111 1111 b	1 1111 1111 b
Mapped Even Addresses	0100 0000 h	1st bank 256Kx64	0 0000 0000 b	0 0000 0000 b
	0100 0008 h	1st bank 256Kx64	0 0000 0000 b	0 0000 0001 b
	0100 1000 h	1st bank 256Kx64	0 0000 0001 b	0 0000 0000 b
	0110 0000 h	2nd bank 256Kx64	0 0000 0000 b	0 0000 0000 b
	011F FFFB h	2nd bank 256Kx64	1 1111 1111 b	1 1111 1111 b
	0120 0000 h	3rd bank 256Kx64	0 0000 0000 b	0 0000 0000 b
	0120 0008 h	3rd bank 256Kx64	0 0000 0000 b	0 0000 0001 b
	0120 1000 h	3rd bank 256Kx64	0 0000 0001 b	0 0000 0000 b
Mapped Odd Addresses	0130 0000 h	4th bank 256Kx64	0 0000 0000 b	0 0000 0000 b
	013F FFFB h	4th bank 256Kx64	1 1111 1111 b	1 1111 1111 b
	0140 0000 h	1st bank 256Kx64	0 0000 0000 b	0 0000 0001 b
	0140 0008 h	1st bank 256Kx64	0 0000 0000 b	0 0000 0011 b
	0140 1000 h	1st bank 256Kx64	0 0000 0001 b	0 0000 0001 b
	0150 0000 h	2nd bank 256Kx64	0 0000 0000 b	0 0000 0001 b
	015F FFFB h	2nd bank 256Kx64	1 1111 1111 b	1 1111 1111 b
	0160 0000 h	3rd bank 256Kx64	0 0000 0000 b	0 0000 0001 b
	0160 0008 h	3rd bank 256Kx64	0 0000 0000 b	0 0000 0011 b
	0160 1000 h	3rd bank 256Kx64	0 0000 0001 b	0 0000 0001 b
	0160 1000 h	4th bank 256Kx64	0 0000 0000 b	0 0000 0001 b
	017F FFFB h	4th bank 256Kx64	1 1111 1111 b	1 1111 1111 b

## Appendix C Programming Examples

---

This appendix contains examples of programming the 82750PD in X86 assembly language and in "C".

### C.1 Host-SFB FIFOs

---

#### C.1.1 Write FIFO

---

The following example of code, shown first in x86 assembly language and then in "C", initializes the 32-bit Host-SFB Write FIFO to address '0xFA0600' and then writes '0x0123' '0x4567' '0x89AB' to that address. The first write to the FIFO control register (WFCON, REG2-BYTE2) has the TEST bit set to initialize the FIFO.

#### *Assembly Language*

---

MOV	DX,REG2BYTE2	.Load the I/O port address into DX
MOV	DX,REG2BYTE2	.Load the I/O port address into DX
MOV	AL,40	.Control Reg. TEST initialization and RS=0
MOV	AH,00	.Least Significant byte of FIFO Address
		.Counter
OUT	DX,AX	.Write the FIFO Address Counter Byte
MOV	AL,01	.Control Reg. RS=1
MOV	AH,06	.Next byte of FIFO Address Counter
OUT	DX,AX	.Write the FIFO Address Counter Byte
MOV	AL,82	.Control Reg. AUTO No TEST and RS=2
MOV	AH,FA	.Most Significant byte of FIFO Address
		.Counter
OUT	DX,AX	.Write the FIFO Address Counter Byte
MOV	DX,REG2BYTE0	.Point to the FIFO Data Registers

## Programming Examples

MOV	AX,0123	First word of data
OUT	DX,AX	Output first two bytes
MOV	AX,4567	Second word of data
OUT	DX,AX	Output next two bytes
MOV	AX,89AB	Third word of data
OUT	DX,AX	Output last two bytes
MOV	DX,REG2BYTE2	Point to the FIFO Control Register
MOV	AX,0083	Will select FIFO Test Register with current operating mode
OUT	DX,AX	Flush FIFO Data Registers

## "C" Language

```
outpw(REG2BYTE2, 0x0040), /* Write FIFO address counter byte *  
                           /* and control (TEST, RS=0) *  
outpw(REG2BYTE2, 0x0601), /* Write next byte of FIFO address  
                           /* counter and control (RS=1) *  
outpw(REG2BYTE2, 0xFA82); /* Write most significant byte of *  
                           /* address */  
                           /* counter and control (AUTO *  
                           /* no TEST, RS=2) */  
outpw(REG2BYTE0, 0x0123), /* Write first word of data *  
outpw(REG2BYTE0, 0x4567); /* Write second word of data *  
outpw(REG2BYTE0, 0x89AB); /* Write third word of data *  
outpw(REG2BYTE2, 0x0083); /* Flush the FIFO */
```

## C.1.2 Read FIFO

The following example of code, shown first in x86 assembly language and then in "C", initializes the 32-bit Host-SFB Read FIFO to address '0x124200' and reads 6 bytes from the SFB. The first write to the FIFO control register (RFCON, REG3-BYTE2) has the TEST bit set to initialize the FIFO. The first write to the 32-bit Read FIFO's FIFO Control Register must have the AUTO bit clear. This code leaves the FIFO in AUTO mode armed to fetch data. (The last OUT instruction sets the 32-bit Read FIFO Request in the hardware.) Because the FIFO

Address Counter is at an even dword address and the FAST bit is set, the FIFO fills with 8 bytes of data with just one internal bus cycle.

### Assembly Language

```

MOV    DX,REG3BYTE2    ;Load the I/O port address into DX
MOV    AL,40            ;Control Reg. TEST and BS=0
MOV    AH,00            ;Least Sig. byte of FIFO Address Counter
OUT     DX,AX           ;Write the FIFO Address Counter Byte
MOV    AL,01            ;Control Reg. BS=1
MOV    AH,42            ;Next byte of FIFO Address Counter
OUT     DX,AX           ;Write the FIFO Address Counter Byte
MOV    AL,92            ;Control Reg. AUTO, FAST, No TEST
                        ;and BS=2
MOV    AH,12            ;Most Sig. byte of FIFO Address Counter
OUT     DX,AX           ;Write the FIFO Address Counter Byte
MOV     DX,REG3BYTE0    ;Point to the FIFO Data Register
IN      AX,DX           ;Input the first two bytes
IN      AX,DX           ;Input the next two bytes
IN      AX,DX           ;Input the last two bytes

```

### "C" Language

```

outpw(REG3BYTE2, 0x0040),    /* Write FIFO address counter */
                           /* byte and control (TEST, RS=0) */
outpw(REG3BYTE2, 0x4201),    /* Write next byte of FIFO */
                           /* address counter and control */
                           /* (RS=1) */
outpw(REG3BYTE2, 0x1292),    /* Write most significant byte of */
                           /* address counter and control */
                           /* (AUTO,FAST, no TEST, RS=2) */
data = inpw(REG3BYTE0),      /* read first two bytes */
data = inpw(REG3BYTE0),      /* read next two bytes */
data = inpw(REG3BYTE0),      /* read last two bytes */

```

## C.2 Indirect I/O to a Configuration Register

The following code illustrates writing to a 82750PD configuration register by the indirect I/O method.

### Assembly Language

```
MOV  DX,REG4BYTE2    ;Load address of high byte of PB
                        ;indirect address
MOV  AL,0FEH          ;PB host CFG register are based at FE0000
OUT  DX,AL            ;init high byte of PB memory
                        ;address
MOV  DX,REG4BYTE0     ;Load address of low & mid byte of indirect
                        ;address
MOV  AX,PBREGNUM       ;desired PB CFG register
OUT  DX,AX            ;init low and mid byte of PB
                        ;memory address
AND  AX,02            ;is it even or odd PB CFG register
JNZ  ODDREG           ;go to odd handler
EVENREG  MOV  DX,REG5BYTE0 ;low 16 bit of data CFG register
MOV  AX,PBDATA        ;data to be written
OUT  DX,AX            ;write data to addressed CFG register
JMP  DONE
ODDREG  MOV  DX,REG5BYTE2 ;high 16 bits of data CFG register
MOV  AX,PBDATA        ;data to be written
OUT  DX,AX            ;write data to addressed CFG
                        ;register
DONE
```

Reading the 82750PD registers would be accomplished in a similar fashion, except that the OUT instructions in the EVENREG and ODDREG section of code would be changed to IN.

### "C" Language

```
outp(REG4BYTE2, 0xFE);    /* Set base address of PB registers */
outpw(REG4BYTE0, PBREGNUM) /* Select desired PB Register */
if(PBREGNUM, 0x02)        /* if the address is odd, write data to */
```

```
outpw(REG5BYTE2, PBDATA); /* REG5BYTE2 */
else                       /* if the address is even write
                           /* data to */
outpw(REG5BYTE0, PBDATA); /* REG5BYTE0 */
```

### C.3 Writing to the *maddr* Register

The 82750PD requires at least one instruction between a write to the *maddr* register and the execution of the instruction that is loaded by the write to *maddr*. The following two examples illustrate inserting this instruction.

### Example 1:

```
maddr = &ADDR1  
jmp ADDR1  
nop  
nop  
ADDR1  
r0 = 0
```

### Example 2:

```
maddr = &INST  
nop  
INST  
r0 = 1
```

## C.4 Writing to the *pc* Register

When a microcode routine writes to the *pc* register, one more instruction is executed before the jump to the new address takes effect. For example

```
pc = &ADDR1  
r0 = r1 jmp ADDR2  
nop  
ADDR1  
r3 = r0
```

## Index

---

82750PB. 1-1. 2-1. 3-1

82750PD interrupt. *See interrupt*

### A

A and B buses. 3-1. 3-3. 3-10

address

core source/destination. 4-6

address bits

82750PD. B-1

DRAM/VRAM. B-1

ALU. 3-4-3-8

operations. 3-6. 4-12

ALU Output Register. *See alu register.*

alu register. 3-4. 4-12

ALUOP field. 4-11

ALUSS field. 4-11

arbiter. 6-1. 6-2-6-4. 6-10. 6-15. 6-49

ASRC field. 4-5

### B

barrel shifter. 3-8

BASE register. 5-10. A-2

base value (B) of code symbol. 3-30

BCNTL register. 5-9. A-2

BDST field. 4-8

bit shifting. 3-3

branch

conditional. 4-2. 4-5. 4-9

unconditional. 4-5

---

Index-1

ATI018019



## Index

---

broadcast mode. *See mode, transmit.*

BSRC field, 4-8

bus

A and B. *See A and B buses.*

internal. *See internal bus.*

bus types, 1-2, 2-2, 5-1

byte swapping, 3-3, 4-9, 4-11, 5-26

## C

c-stat register, 3-36

cc register, 3-3, 3-4-3-6, 3-11, 4-3, A-3

bit definitions, 3-6

ccontrol register, 3-48, 3-49-3-50, 3-54, 5-57, 5-61, A-4

byte offset, 3-60

CFGBASE register, A-5

CFGBUS register, A-5

CFGDATA register, A-6

CFGGEN register, A-6

CFGINT register, A-7

CFGNUM register, A-8

CFGSFBI4 register, A-9

CFGSFBI5 register, A-10

CFSEL field, 4-3

cintflag register, 3-48, 3-50, 3-54, 5-61, A-11

byte offset, 3-60

circbuf register, 3-15

and Core Output FIFO, 3-20

and Statistical Decoder, 3-37

Circular Buffer Register. *See circbuf register.*

clock

internal, 3-1

- SFBI. 3-1
- SynchroLink. 6-49, 6-50
- CMD0 register. A-11
- CNT field (bit). 4-8
- cnt register. 4-8, 4-13
- cnt2 register. 4-8, 4-13
- code
  - fixed-length. 3-24
  - variable-length. 3-24
- code description table. 3-25, 3-27-3-30, 3-37
  - reading. 3-36
  - writing. 3-35
- Condition Code Register. *See* cc register.
- condition flag. 4-2, 4-3, 4-5, 4-8
- condition flag outputs. 3-4
- core. 1-1
  - architecture. 3-1
  - overview. 3-1
- Core Control Register. *See* ccontrol register.
- Core Input FIFO. 3-14-3-19
  - initialization. 3-18
  - registers. 3-15
- Core Interrupt Flag Register. *See* cunflag register.
- Core Output FIFO. 3-19-3-62
  - initialization. 3-22
  - registers. 3-20
- core registers. 0-1-7
  - access by host CPU. 5-3
  - access to. 3-2
  - core access to. 3-55
  - host access to. 3-55
  - memory address offsets. 3-56
  - naming. 1-3
  - source/destination addresses. 3-56

## Index

---

Core Status Register. *See cstatus register.*

cstatus register. 3-48, 3-52, 3-54, 5-61, A-12  
byte offset. 3-60

## D

data holding register. *See* Host-SFB Read FIFO, Host-SFB Write FIFO

data packet. 6-1, 6-4-6-16, 6-20, 6-33  
service completion. 6-11, 6-12  
service request. 6-9, 6-10

data RAM. *See* DRAM.

decoder. 3-33

Decoder Hold Read Register. *See \*stat# register.*

Decoder Read Register. *See \*stat-c register.*

Decoder SFB Pointer Registers. *See stat-hi, stat-lo registers.*

Decoder Table Write Register. *See stat-ram register.*

decoding algorithm. 3-30

Device 0

82750PD ineligible. 5-6  
role in configuration. 5-5

device ID

on SFBI. 5-5

direct I/O. 5-1, 5-3, 5-13

display memory. 2-9

DRAM. 2-3, 3-9-3-10

DRAM/VRAM. B-1

dramN register. 3-9

## E

EISA bus. 5-1

POS registers. 5-9

EMSCFGn register. A-13

"EMS" addressing. 2-6

"EMS" mode. 5-17-5-24

access to memory address space. 5-2

configuration registers. 5-19, 5-21

enabling. 5-22

mapping. 5-18

PAR registers. 5-23

setting up. 5-20

window. 5-18

window, enable/disable. 5-7

event

SFBI. 6-22, 6-52

## F

fcnt register. 3-14, 6-52

and meta-interrupt. 5-57

field

DATA. 6-7, 6-21, 6-23, 6-33

data packet. 6-6

FCODE. 6-6, 6-9, 6-21, 6-33, 6-40

FMATCH. 6-21

microcode instruction. 4-2-4-13

service number. 6-7, 6-40

Field Counter. *See* fcnt register.

FIFOs

core FIFOs. *See* Core Input FIFOs, Core Output FIFOs

Host-SFB, *see* Host-SFB FIFOs

flush

Core Output FIFO. 3-23

frozen (processor). 3-13, 3-19, 3-23, 4-3

function code. 6-6, 6-8-6-17, 6-20, 6-21, 6-30, 6-32

service completion. 6-7, 6-8

service request. 6-6, 6-8

## G

GENCON register. 5-58, 5-59, 5-61, A-13

## Index

---

General Control Register. *See* GENCON register.

General Status Register. *See* GENSTAT register.

GENSTAT register, 5-60, 5-61, A-14

## H

Halt mode, 3-13, 3-49

conditions for, 3-52

Horizontal Line Counter. *See* lcnt register.

host bus. *See also* UHBI.

address configuration, 5-6

bus types, 5-1

host I/O registers, 5-2, 5-13

access by host CPU, 5-3

address computation, 5-17

address format, 5-16

address map, 5-13

Host-SFB FIFOs, 2-2

Host-SFB Read FIFO, 5-2, 5-42-5-54, 5-45

automatic increment, 5-48

data holding register, 5-44

empty, 5-45

full, 5-46, 5-47

prevent bus acquisition for testing, 5-47

programming example, C-2

registers (REG 3), 5-14

residual data, 5-47

saving and restoring the state, 5-53

Host-SFB Read FIFO Address Counter Register. *See* RFCNTR register.

Host-SFB Read FIFO Control Register. *See* RFCON register.

Host-SFB Read FIFO Data Register. *See* RFDATA register.

Host-SFB Read FIFO Selected Byte Register. *See* RFSELB register.

Host-SFB Read FIFO Test Register. *See* *RFTEST* register.

Host-SFB Write FIFO. 5-2. 5-28-5-42  
automatic increment. 5-36  
data holding register. 5-32. 5-38  
empty. 5-31. 5-33  
flush. 5-32  
full. 5-34  
prevent bus acquisition for testing. 5-35  
programming example. C-1  
registers (REG 2). 5-14  
saving and restoring the state. 5-40  
WFCNTR register.. *See* *WFCNTR* register.

Host-SFB Write FIFO Address Counter. *See* *WFCNTR* register

Host-SFB Write FIFO Control Register. *See* *WFCNTR* register.

Host-SFB Write FIFO Data Register. *See* *WFDATA* register.

Host-SFB Write FIFO Selected Byte Register. *See* *WFSELB* register.

Host-SFB Write FIFO Test Register. *See* *WFTEST* register.

I/O. *See* *direct I/O*, *indirect I/O*.

I/O window  
configuration. 5-5. 5-7  
enable/disable. 5-7

ID  
message. 6-6. 6-10. 6-12. 6-15. 6-51

illegal  
bit combinations. 1-4

INDADDR register. A-16

INDDATA register. A-16

## Index

---

- indirect address register (REG4). 5-14, 5-24
- indirect data register (REG5). 5-14, 5-25
- indirect I/O. 5-24-5-27
  - access to memory address space. 5-1
  - programming example. C-4
- inN-c register. A-14
  - bit definitions. 3-15
  - reading when empty. 3-19
- inN-lo, inN-hi registers. 3-15
- \*inN register. 3-15
- INT CFG register. 5-61
- interleaved memory. 2-6, 2-8, 2-9
- internal bus. 0-2-8, 3-1
  - access by host CPU and 82750 core. 5-3
  - and Core Input FIFO. 3-14
  - consecutive cycles by Host-SFB Read FIFO. 5-46
  - consecutive cycles by Host-SFB Write FIFO. 5-34
- internal bus cycles
  - single and multiple. 5-4
- interrupt. 3-47, 5-53-5-62, 5-54
  - configuring. 5-62
  - core registers. 3-47
  - DINT bit for edge- or level-triggered host bus. 5-59
  - enable bits. 3-50
  - enable bits, status of. 3-52
  - Even Field interrupt. 3-50, 3-54
  - flags. 3-50
  - handler. 5-60, 5-62
  - HLINE code. 5-58
  - microcode interrupt. 3-8, 3-50
  - Odd Field interrupt. 3-50, 3-54
  - overflow flag. 3-51
  - set-up procedure. 5-61

sources. 3-47  
summary. 3-54  
to host. 3-47, 5-2, 5-54  
VBUS code interrupt. 3-54  
Vertical Blanking interrupt. 3-50, 3-54  
VEVEN code. 5-58  
VODD code. 5-58

"interrupt host" operation. 3-8

invitation acknowledge. 6-4, 6-5, 6-6, 6-7, 6-10, 6-12, 6-15

IO\_BASE. A-2

IO\_START. 5-13, A-2

ISA bus. 5-1

POS information. 5-9

ISA bus type. 3-50

## L

latch

A bus. 4-11

ALU input. 4-11

ALU output. 4-11

B bus. 4-11

LC field (bit). 4-13

lcnt register. 3-14, 6-52

and meta-interrupt. 5-57

LIT field. 4-9

logical address. 1-2

loop counter. 4-13

loop counters. 3-10-3-11

loopback capability. 6-4

## M

maddr register. 3-12

programming example. C-5

writing to. 3-13



- Match Service Register. *See* *MSRV* register.
- MCFG register. 5-56, 6-18, 6-47, 6-48, A-17
- mcodeN register. 3-12, 3-13
- memory
  - DRAM. *See* *DRAM*.
  - interleaved. *See* *interleaved memory*.
  - MRAM. *See* *MRAM*.
  - non-symmetrical. B-1
  - off-screen. *See* *off-screen memory*.
  - Reserved. *See* *Reserved memory*
  - SFB. . *See* *SFB*.
  - symmetrical. B-1
- memory address space. 2-4-2-7
  - SynchroLink registers. 6-17
- message
  - broadcast. 6-21, 6-30-6-32
  - service completion. 6-11, 6-21, 6-22, 6-35-6-43, 6-36, 6-39, 6-40
  - service completion programming sequence. 6-38, 6-42
  - service completion, defined. 6-40
  - service request. 6-5, 6-9, 6-21, 6-22, 6-35-6-43, 6-37, 6-39
  - service request acknowledgement. 6-9
  - service request programming sequence. 6-38, 6-42
- Message Configuration Register. *See* *MCFG* register.
- Message Status Register. *See* *MSTATUS* register.
- meta-interrupt. 3-48, 5-53-5-62, 5-54, 6-1, 6-3, 6-22, 6-35, 6-50, 6-52-6-54
  - effect on fcnt register, lcnt register. 5-57
  - HLINE code. 5-57
  - V1CODE or V2CODE. 6-53
  - VEVEN code. 5-57
  - VODD code. 5-57
- Micro Channel bus. 5-1
  - POS registers. 5-10

Micro Channel POS3 Register. A-18, A-19

microcode

- instruction, Chapter 4
- instruction format, 4-1
- instruction sequencing, 4-1
- instructions, 1-1

Microcode Address Register. *See maddr register.*

Microcode Instruction Register. *See mcodeN register.*

microcode interrupt. *See interrupt, microcode.*

Microcode RAM. *See MRAM.*

mode

- broadcast, 6-23
- broadcast receive, 6-20, 6-24, 6-32, 6-44
- broadcast transmit, 6-20, 6-24, 6-30, 6-42
- service, 6-23
- service completion, 6-24
- service request, 6-24
- XCVRn register, 6-23, 6-24, 6-51

modes

- Pixel Interpolator. *See Pixel Interpolator modes.*

MRAM, 2-3, 3-11-3-14

- organization, 4-2
- reading, 3-12
- writing instruction to, 3-12

MSRV register, 6-7, 6-18, 6-45, A-19

MSTATUS register, 6-18, 6-19, 6-43, A-21

N

NACK, 6-21, 6-36

NADDR field, 4-2, 4-4, 4-5

notation, 1-3

O

off-screen memory, 2-9

operating modes, 3-49

## Index

---

outN++ register. 3-20  
outN-c register. 3-20, A-22  
outN-lo, outN-hi register. 3-20  
\*outN register. 3-20

## **P**

PAR registers. 5-13, 5-18  
PARn register. A-23  
pc register. 3-12, 4-2  
    programming example. C-6  
    writing to. 4-2  
PCI bus. 5-1  
    POS registers. 5-11  
PCI bus type. 3-50  
performance monitor operation. 3-8  
physical address. 1-2  
Pixel Interpolator. 3-41-3-47  
    modes. 3-42  
    pairing. 3-42  
    pipelining. 3-43, 3-46  
    pixel pair phase. 3-43, 3-46  
    pixel pairing selection. 3-45  
    start up sequence. 3-43  
    start-up delay. 3-43, 3-44, 3-47  
    weightings. 3-41, 3-44  
Pixel Interpolator Control Register. *See pixant-c register.*  
Pixel Interpolator modes  
    Random-2D. 3-42, 3-43-3-44  
    selection. 3-45  
    Sequential-2D. 3-42, 3-43  
pixant-c register. 3-44, A-24  
pointer  
    DRAM. *See dramN register*

SFB. *See* SFB pointer.

polarity

bits in run-in sequence. 3-37

POS registers. 5-6, 5-7

EISA bus. 5-9

ISA bus. 5-9

Micro Channel bus. 5-10

PCI bus. 5-11

VL-Bus. 5-12

priority

transmission requests by XCVRn registers. 6-29

Program Counter. *See* pc register.

programming

examples. 1-2

programming sequence

broadcast receive mode. 6-33

broadcast transmit mode. 6-31

protocol

devices in logical slot. 5-5

## R

R (index to code description table). 3-27

r0 register. 3-3, 4-10

Receive Service Register. *See* RSRV register.

references. 1-5

REG0-REG7. 5-14

registers

core. 3-3

general purpose. 3-3-3-4

naming. 1-3

RAM. *See* MRAM registers

SynchroLink interface. 5-54, 6-17, 6-53

Reserved

bits, bytes. 1-3

Reserved memory, 2-10

RFCNTR register, 5-42, 5-43, 5-49-5-50, A-25  
    accessing bytes, 5-44  
    warning on write, 5-46

RFCON register, 5-42, A-26  
    warning on write, 5-46

RFDATA register, 5-42, 5-49, 5-52-5-53, A-25  
    reading, 5-50

RFSELB register, 5-42, 5-44, A-29

RFTEST register, 5-45, 5-50, A-29

RSRV register, 6-18, 6-39, 6-43, 6-44, 6-45, A-30  
    affected by meta-interrupt event, 5-55  
    effect of SFB event, 6-54

RSVD  
    bits, bytes, 1-3

run-in sequence, 3-26

S

service acknowledgement, 6-5, 6-11

service completion. *See message, service completion; see mode, service completion.*

service number, 6-7, 6-8, 6-9, 6-10, 6-11, 6-12, 6-21, 6-40, 6-47  
    recognized, 6-41, 6-45, 6-47

service request. *See message, service request; see mode, service request.*

service successful, 6-13

SFB, 1-1, 0-1-10  
    access, 5-3  
    access by host CPU, 5-2  
    and Core Input FIFO, 3-14  
    and Core Output FIFO, 3-19

SFB pointers  
    byte addresses, 3-57, 3-60

for Statistical Decoder, 3-38

SFBI, 1-2, 0-1-3, 6-1  
event, 5-54

SFBI event, 5-55, 6-1, 6-22

Shared Frame Buffer Interconnect. . See *SFBI*.

SHFT field, 4-9

signal event, 6-4, 6-7, 6-12, 6-15

slot (logical)  
configuration, 5-5  
protocol for devices sharing, 5-5  
shared by multiple devices, 5-5

snooping (data), 5-6  
Micro Channel bus, 5-11  
PCI bus, 5-12

source  
on SynchroLink, 6-2-6-4

source/destination coding, 4-5, 4-6

start bit, 6-4, 6-6, 6-10, 6-15

stat-c register, 3-25, A-31  
bit definitions, 3-34

stat-hi, stat-lo registers, 3-25

stat-ram register, 3-25, 3-36

\*stat register, 3-25, 3-36

\*stat# register, 3-25, 3-37

Statistical Decoder, 3-24  
codes, 3-25  
decode time, 3-38  
END mode, 3-31-3-32  
registers, 3-24  
SHORT mode, 3-30-3-31, 3-33, 3-37  
variable-length code, 3-25

Statistical Decoder Control Register. See *stat-c*  
register.

## Index

---

- Step mode. 3-49, 4-3
- switch over. 6-4, 6-6, 6-7, 6-9, 6-10, 6-12, 6-15
- SynchroLink. 1-2, 0-2-3, 2-2, 6-1
  - frequency. 6-48
- SynchroLink interface. 0-3, 3-1, 6-1, 6-22
  - affected by meta-interrupts. 5-55, 6-54
  - configuration. 6-47, 6-51
  - registers. *See registers. SynchroLink interface.*
- SynchroLink registers. 0-3-4
  - access by host CPU. 5-3
- T
- target device. 6-7
  - on SynchroLink. 6-2-6-4
- Transceiver Register. *See XCVRn register.*
- U
- UHBI. 1-2, 0-2-4, 2-2
  - access to multimedia system. 5-1
  - address configuration. 5-4
  - bus types supported. 5-1
- Universal Host Bus Interface. *See UHBI.*
- V
- V1CODE. 6-50
- V1CODE event. 5-56
- V2CODE. 6-50
- V2CODE event. 5-56
- variables
  - notation for. 1-3
- VBUS. 3-1, 5-57, 6-1
  - code. 5-56, 5-57
  - priority. 5-57
- VBUS code. 3-14
  - updating lcnt register. 3-14
  - VEVEN code. 3-14

VODD code. 3-14

VBUSIN pins. 6-50

VL-Bus. 5-1

POS information. 5-12

## W

wait states

host bus reading from Host-SFB Read FIFO. 5-46

host bus writing to Host-SFB Write FIFO. 5-35

WFCNTR register. 5-29, 5-36, A-32

accessing bytes. 5-36

warning on write. 5-37

WFCON register. 5-29, 5-31-5-36, A-33

byte selection. 5-31

WFDATA register. 5-28, 5-37, A-34

alternation of bytes for write. 5-38

loading. 5-38

loading procedure. 5-39

write restrictions. 5-37

WFSELB register. 5-31, 5-33, 5-41, A-35

WFTTEST register. 5-33, 5-41, A-35

workspace

82750PD. 2-10

graphics. 2-10

## X

XCVRn register. 6-18, 6-20, 6-37-6-39, A-36

broadcast receive mode. 6-32

broadcast transmit mode. 6-30

enabling meta-interrupts. 5-55, 6-54

mode. 6-23



UNITED STATES INTERNATIONAL TRADE COMMISSION  
Washington, D.C. 20436

Before The Honorable Debra Morriss  
Administrative Law Judge



\_\_\_\_\_  
In the Matter of )  
 )  
 )

CERTAIN VIDEO GRAPHICS DISPLAY )  
CONTROLLERS AND PRODUCTS )  
CONTAINING SAME )  
\_\_\_\_\_ )

Inv. No. 337-TA-412

**EXPERT REPORT OF WILLIAM G. MEARS**

UNITED STATES INTERNATIONAL TRADE COMMISSION  
Washington, D.C. 20436

Before The Honorable Debra Morriss  
Administrative Law Judge

\_\_\_\_\_  
In the Matter of )

CERTAIN VIDEO GRAPHICS DISPLAY )  
CONTROLLERS AND PRODUCTS )  
CONTAINING SAME )  
\_\_\_\_\_ )

Inv. No. 337-TA-412

**EXPERT REPORT OF WILLIAM G. MEARS**

I have prepared this expert report at the request of respondent ATI Technologies, Inc. and if called to testify as the contents of the report, could and would testify competently thereto. In this report, I consider issues of validity relating to asserted claims of U.S. Patent No. 5,598,525 ("the '525 patent"). I will describe the 1280 and VIPER series graphics and video controller products that were designed, manufactured, and sold by Parallax Graphics, Inc., a company that I co-founded in 1982. As I detail in my report, the 1280 and VIPER graphics and video controllers contain features and functionality that are described and claimed in the '525 patent. In fact, we at Parallax were implementing the architecture and functionality set forth in the '525 patent many years before the patent. We were years ahead of most others in the industry in our work dealing with the merger of video and graphics.

As fully explained below, I believe that asserted claims of the '525 patent are invalid in view of the 1280/VIPER products. In forming my opinions, I rely on my knowledge and

experience in the field of graphics and video controllers, and on certain documents and information that are specifically referenced in the report. I would like to point out that my work in this case is continuing and that this report represents only a current evaluation of my positions on validity. I may supplement this report as additional information becomes available and my trial testimony may also include additional views developed in connection with my ongoing work in this case. I also may submit a rebuttal expert report regarding any issues raised by Cirrus's experts.

### **MY GENERAL BACKGROUND**

I have over 15 years of experience in video and graphics product development. That experience is summarized in my resume which is attached hereto as Exhibit 1. At the hearing in this case, I may offer testimony relating to my background and experience, some of which is captured in my resume.

Briefly, however, much of my experience derives from my work at Parallax from 1982 through 1991, a company I co-founded. I describe this experience in more detail below. After Parallax, I worked at Force Computers as a manager in the VME Engineering group where I designed and developed VME processor cards that were successfully launched into the market. In early 1994, I started my own contracting business where I worked with a number of companies to develop integrated Computer Graphics/Video products for a variety of applications. In early 1995, I was asked to be Director of R&D at Viewgraphics, Inc. where I developed and brought to market the company's Serial Digital Adapter (SDA) & Digital Data Adaptor (DDA) product families. These products successfully interfaced

broadcast quality real-time video and were successfully introduced to the computer market. I am now Vice-President of Cogent Technology, a company that I have co-founded. At Cogent, I have developed architecture and performed designed work for a MPEG-2 transport stream processing product line. This product line is intended to facilitate the deployment of the digital television broadcast infrastructure. I have a Bachelor of Science in Electrical Engineering from Cornell University.

I am being compensated for my efforts in this case at my standard consulting rate of \$250/hr. My compensation in this case is not tied to the result of the litigation.

In the preceding four years, I have not provided any expert testimony at trial or in deposition. In the preceding ten years, to the best of my recollection, I have authored no publications dealing with the subject of graphics and video.

#### **PARALLAX GRAPHICS - THE COMPANY**

In 1982, I co-founded Parallax to fill a need in the market for high performance and low cost graphics. We envisioned designing and producing boards that would replace high overhead and high cost "box" like equipment, yet provide at least the same level of functionality and performance of previous generation products. Because of my previous exposure to video, I felt it was critical to integrate video and graphics in the architecture of our products.

In 1982, we designed our first product referred to as the Parallax 600. This product had the ability to capture and display video in real-time and could overlay graphics on a live video background using a simple form of color-keying. This product was successful

predominantly because of its ability to integrate graphics and video. We sold many hundreds of these systems which were used by, for example, Clairol to do visual make-overs of people's images by capturing real-time images and overlaying them with graphical data. The product also captured real-time motion video for overlay. Although the product was successful, it had problems. Specifically, the Parallax 600 processed video in the graphics space. That is, the product used 8 bits per pixel to store both graphics and video data. The color was derived by using RGB format with three bits for red, three for green, and two for blue. This only provided for a total of 256 colors that were used for both video and graphics. This was not enough color subtlety for high quality video.

So we began to explore alternative storage formats for the video data. The human eye sees most detail in terms of black/white images or luminance. Thus, if more bits of storage are allocated to "luma" image and less to the "chroma" image, the resulting data format would be more visually appealing with the same number of bits. This storage data format is well known as YUV in the video industry (and indeed was known in the 1950's during the transition from B/W to color television broadcast). When the data is sampled in the YUV format, only one sample of U & V (or chroma) is needed for each four Y (or luma) samples. For example, across a sample of four pixels there would be four 6 bit luma samples, and one sample of each U & V at a resolution of four bits. The visual perception created a pixel of 16.536 possible different colors, far more than the 256 colors of the previous system. Also, this approach freed the color tables to be available to the user's preference for graphics colors.

We (and, in particular, I) applied these principles to the design and architecture of a next generation product referred to as the 1280. Design for this product began by early 1984. In designing the 1280, we had several design goals in mind: (1) display resolution at 1280 x 1024; (2) ability to capture and display with enhanced resolution as compared to the 600; (3) increased graphics performance with increased off-screen memory; (4) and increased integration to reduce product size and power demands. With these objectives, we designed an architecture for the 1280.

The design of the new video encoding format, as well as the ability to display video on a high resolution display was innovative and deemed patentable at the time. However, Parallax Graphics chose to use protection of trade secrets as our operating paradigm, rather than pursue patents for the purpose of protection. As described below, however, there are contemporaneous documents describing the operation of our products.

During this time frame (early to mid 1984), we also had discussions with Martin Marietta about implementing our design and architecture in conjunction with products produced by them. In this regard, Martin Marietta actually participated in the design specifications for the 1280 in order to ensure that our product would meet their needs. Martin Marietta was developing a portable tactical computer that used video discs to store maps, a product referred to as ASAS. This product was being designed for the military. The 1280 was to be the display for the ASAS work station, and was required to capture live video, and overlay graphics onto the video. Because of their need to display graphics over video, we

enhanced our design to include "graphics-over-video" features. These enhancements included the ability to draw graphics directly over the video, in addition to having video windows.

The result of our efforts were completed sometime in 1985 and in 1986 we were shipping the product to customers, including Martin Marietta. A write up of our product appeared in a March 25, 1986 article entitled "Coprocessors Provide Integrated Video and Graphics," written by Marty Picco, a co-founder of Parallax.

The 1280 series graphics processor was very successful, particularly because of its unique ability to capture and display live video. For example, Boeing used it to dramatically improve its documentation for its 747. Boeing used the 1280 to create laser disc based manuals that replaced the paper manuals previously used at Boeing, which filled a tractor-trailer. The system was implemented by Boeing to display textual description of procedures relating to its many thousands of parts and would display an instructional video along side graphics and text. Our controllers were also used to pilot remote control robots used in the clean-up of the Three Mile Island nuclear accident. To the best of my recollection, other customers include Electronic Data Systems, the Israeli Ministry of Defense, and Texas Instruments. The 1280 was designed for use in Q-Bus and VME workstations from companies like Digital Equipment Corp., and Sun; there was also a version for IBM PCS. Through the introduction of this product, our annual sales volume increased from relatively little to over three million dollars per year. The products, depending on the configuration, sold from \$9,000 to \$35,000 each.

09374041-092899  
The 1280 was successful and possessed many product features not otherwise available in the market place during this period of time. In an effort to keep the 1280 product competitive in the market place, we envisioned a next generation product called the VIPER. This product would reduce the size and cost of the 1280 by consolidating discrete circuitry into gate arrays that reduced the power and space requirements in the design. The VIPER was introduced sometime in 1988. The VIPER is described in a version of its users manual dated 1989 (ATI031566-ATI032067). A picture of the VIPER displaying graphics and multiple video windows is depicted in a 1989 publication entitled The NEWS Book. We were selling an average of 50 VIPER controllers per month during this time period and our annual revenues increased to 9 to 10 million dollars per year. I note that at this period of time, the market began to see the emergence of windowing systems such as NEWS for Sun workstations and "X" for DEC workstations. Our product was used with both which further enhanced our sales.

The success of the Parallax 1280 and VIPER products did not go unnoticed in the industry. In 1989, we were acquired by a company called Dynatech Corporation, a 400 million dollar publicly traded company, which continued to sell the product. I left the company in 1991.

### **PARALLAX - THE PRODUCTS**

The Parallax 1280/VIPER series graphics and video processors can be understood with reference to a Parallax 1280 technical manual (ATI032068-ATI032775); a Parallax VIPER technical manual (ATI031566-ATI032067); an excerpt from The News Book at pp. 178-221



(ATI0\_\_\_\_-ATI0\_\_\_\_); and a March 25, 1986 Digital Design article entitled "Coprocessors Provide Integrated Video and Graphics." by Marty Picco (a co-founder of Parallax). I also have in my possession sample 1280 and VIPER boards.

#### A. THE 1280 SERIES

The architecture and functionality of the 1280 series controller is now described. The architecture can best be understood by considering its different functional blocks. In the Parallax 1280 technical manual (ATI032068-ATI032775), there is a high level functional block diagram that generally depicts the functionality of the 1280 product. See Figure H-5 on page H-27 (ATI032738). For purposes of this discussion, I will consider the following functional blocks: (1) control/processor section (2) display memory section; (3) video input section; and (4) display generation section.

The control processor section consisted of a core processor that was responsible for choreographing all operations of the controller. The core processor accepted high level instructions from the host and translated them into multiple low-level read and write instructions that were issued to the display memory circuitry. The host sent both graphics and video data to the core processor over the system bus. More specifically, the graphics or video data from the host was received by the Bus Interface Unit (BIU), which is a multi-aperture port. As well as converting host commands, the core processor was also responsible for controlling the display generation circuitry. Additionally, the core processor also directed the read out of data from memory.

The core processor performed various manipulations on display memory which included (1) writing a single pixel; (2) writing a horizontal run of solid color pixels of programmable length and position; (3) writing a repeating pattern of pixels (STIPPLE); (4) copying a horizontal run of pixels from one area in memory (such as off-screen) to another; (5) and importing horizontal run from the real-time video interface. As well as controlling what pixels were written, the core processor also had control over the Access Attribute Control, which had the role of modifying and enabling the writing of each pixel on an individual basis according to the mapping specified by the Access Attribute Control.

In the 1280, display memory was interleaved by a factor of eight. This means that eight locations are read or written simultaneously. The interleave was organized horizontally such that a single access to memory would read eight horizontally consecutive pixels. This interleaving was important to achieve the necessary data rates demanded by the display while still retaining sufficient bandwidth for drawing into and modifying display memory. Access to display memory was timed-division-multiplexed, meaning that either the display circuitry or the access circuitry would have sole access that was traded off as necessary. The display circuitry, however, always had priority.

The display memory (or frame buffer) of the 1280 had a dimension of 2048 x 2048, or 320 percent of the memory necessary for a 1280 x 1024 display. The frame buffer had on-screen and off-screen memory that each stored any kind of data including video and graphics data. The frame buffer also stored the graphics and video data in their respective graphics and video formats (i.e. RGB for graphics and YUV for video). The frame buffer had no

restriction on the size or position of the video or graphics areas (other than video areas needed to start and end on multiples of 4 pixels horizontally). While the standard configuration of the product employed an 8 bit depth of the display memory, we also offered 16 and 24 bit versions of the product to certain customers who demanded it. With this extended memory option, certain additional display features were made available.

The 1280 also had a real-time video port as shown in the Figure. The function of this port was to decode an analog RS170 color video signal and convert it to a digital bit stream. Synchronization information from the analog signal was separated from image data and sent to the control processor that managed the frame grabbing process. The control processor also included circuitry that generated a memory address for the video data. The image data was formatted into video mode byte stream that was written directly into the display memory during the frame grabbing process.

Finally, the display generation circuitry of the 1280 is described. This circuitry performed two basic functions: (1) control the timing and synchronization of memory retrieval from the frame buffer to the raster scan of the display monitor and (2) process the stream of pixels as either video format pixels or graphics, and deliver them to the monitor at the appropriate time.

The timing and synchronization circuitry used an origin register that was written by the core processor. The origin register included the number of lines of data to be displayed from that origin. When the display of those lines were completed, the display generator would interrupt the core processor for the next display segment. This technique permitted the

screen to be split into multiple independent display regions. As well as controlling split screen operations, the control processor programmed a SYNC generator with appropriate monitor parameters.

While only 1280 x 1024 pixels were displayable in any given frame, ALL of the display memory was viewable by simply changing the values loaded into the display origin register. This means that graphics and video format data stored in off-screen memory could be easily displayed without copying it to the on-screen area of memory.

The second part of the display circuitry processed the stream of pixels. Pixels were processed as either video or graphics depending on the value of the display attribute map (DAM). This extra bit plane in the frame buffer was used to distinguish areas of video format data from areas of graphics format data. The display stream hardware was comprised of two pipelines: one pipeline for video and one pipeline for graphics. The graphics pipeline passed each pixel through an 8 x 24 color look up table yielding a 24 bit RGB pixel. The video pipeline processed the video format data, which included decoding YUV into RGB.

A multiplexer served as data selector, or overlay generator, that would selectively pass data from either the video display pipeline or the graphics display pipeline through to the monitor in a manner that was mode dependent, i.e. graphics mode, video mode, and graphics over video mode. All display memory data was passed to both pipelines; based on the DAM bit, contents from the display memory would be output as either video format data, or as graphics format data. Data output from the other pipeline for that pixel location was simply discarded. In an optional display mode known as graphics over video, bit zero of each

luma sample in the video pipeline acted like a color key. In the case where bit zero was off (i.e. 0), video samples passed through the video pipeline as described above. In the case where bit zero was on (i.e. 1), bits 1-5 were treated as a pseudo color graphics pixel and displayed from the graphics pipeline. Thus, 32 colors were made available to display graphics pixels within a video region on a pixel by pixel basis.

In the case of 16 and 24 bit memory options, the display circuitry could treat graphics areas as true color pixels. This means that if the DAM bit was a 0, that the graphics pixel would be generated in true color mode from a full 16 or 24 bits of display memory. If the DAM bit was a 1, then the video format pixel would be generated exactly as described above.

## **B. THE VIPER SERIES**

As I noted above, we at Parallax created the next generation product called the VIPER in order to reduce power consumption, space requirements, and the cost of the processors to our customers. This was intended to ensure that the 1280 had continued viability into the future. With these design objectives in mind, we modified the 1280 by consolidating discrete circuitry into gate arrays that reduced the power and space requirements in the design. The functionality and architecture of the VIPER is otherwise the same as the 1280. As such, everything I said above with respect to the 1280 equally applies to the VIPER.

## **C. SUMMARY OF FUNCTIONALITY FOR THE PARALLAX 1280/VIPER PRODUCTS**

To summarize the architectural and functional features of the 1280/VIPER, I identify the attributes of the 1280/VIPER products:

- (1) The 1280/VIPER merged video and graphics data from a single multi-format

frame-buffer for simultaneous display on a computer monitor. Multiple video windows could appear anywhere on the display monitor. There was no specific correspondence between the location of the video data stored in the frame buffer and the location of the video window on the monitor.

(2) The 1280/VIPER had a multi-aperture port for receiving both video and graphics data from the system bus. The arriving data had a host assigned address that was temporarily stored in an address buffer.

(3) The 1280/VIPER used a single frame buffer to store both video and graphics data in their native formats (i.e. RGB and pseudo-color for graphics and YUV for video). The frame buffer had on-screen and off-screen areas. Also, YUV video data could be stored in either off-screen or on-screen memory and RGB graphics data could be stored in either on-screen and off-screen memory.

(4) The 1280/VIPER contained circuitry for writing into on-screen and off-screen areas of the frame buffer. The 1280/VIPER also contained circuitry for selectively retrieving data from on-screen and off-screen portions of the memory and directing the data to the back-end graphics and video pipelines.

(5) The 1280/VIPER had a graphics over video mode that utilized a version of color keying to overlay video with graphics. In other modes, the control overlay circuitry switched data streams at the output with the presence of the DAM bit.

(6) The 1280/VIPER has a real-time video port that decoded analog RS170 color video signal and converted it to a digital bit stream in a YUV format. The control processor

also included circuitry that generated a memory address for the video data. The image data was formatted into a video format byte stream which was written directly into the display memory during the frame grabbing process.

### **THE '525 PATENT**

The '525 patent is entitled "Apparatus, System and Method For Controlling Graphics And Video Data in Multimedia Data Processing And Display Systems," and issued on January 28, 1997. The '525 patent contains a total of 47 claims with claims 1, 13, 25, 34, 37, and 43 being independent claims. In the hearing in this case, I may offer testimony generally describing the '525 patent from the perspective of one skilled in the art.

I understand that a patent claim is invalid based on anticipation if one prior art reference or product includes all the limitations of that claim. I also understand that a patent claim is invalid based on obviousness in view of one or more prior art references. I am told that when examining the question of obviousness, one must consider the following factors: scope and content of the prior art; level of skill in the art; differences, if any, between the invention claimed and the prior art; secondary considerations including commercial success, copying, long-felt need, and other independent development of the claimed invention. I am also told that obviousness must be tested as of the time the invention was made. One must ask the questions, "Would this have been obvious to a person having ordinary skill in the art at the time the invention was made?" I am also told that the test for obviousness is what the combined teachings of the references would have suggested, disclosed, or taught to one of ordinary skill in the art.

158730-1104260

I also understand that the plaintiff in this case claims to have made the invention of the '525 patent in September 1993. At that time, the level of skill in the art of graphics controller development and also the art of video product development was quite high. At the same time, the multimedia market was expanding quite quickly, creating an incentive to provide products which combined video with graphics. There was a large demand for designers with overlapping skills. A person of ordinary skill in this combined area would have experience designing graphics controllers as well as video processing circuits and would have been familiar with many different design choices. I was such a designer, and at the time, knew other designer with similar skills in the art of video/graphics controllers. In the following chart, I compare the elements of the claims of the '525 patent with the Parallax 1280/VIPER products and with my knowledge of ordinary skill in the graphics/video art. In doing so, I interpret the words of the claims according to their ordinary meanings to me as an engineer. I am not a lawyer and am not attempting to give legal meaning to the claims. I understand that doing so requires consideration of the patent specification, the patent prosecution history, and numerous legal principles. I have read the patent specification but not the prosecution history. Once the claims have been legally construed in this case, I reserve the right to modify the chart below. I also must again stress that my investigation is continuing, and that additions and/or deletions may be made in the future and reflected in my trial testimony. For trial, I may also prepare diagrams, other charts, and possibly a demonstration that illustrates the architecture and operation of the Parallax products.



## APPLICATION OF THE PARALLAX 1280/VIPER TO THE ASSERTED CLAIMS

### **CLAIM 37 IS INVALID OVER PARALLAX**

CLAIM 37	PRIOR ART
37. A display controller comprising:	Parallax 1280/VIPER
circuitry for selectively retrieving data from an associated multi-format frame buffer for simultaneously storing graphics and video data;	circuitry in Parallax 1280/VIPER for selectively retrieving graphics (RGB) and video (YUV) data from display memory.
a first pipeline for processing words of graphics data selectively retrieved from said frame buffer; and	circuitry in Parallax 1280/VIPER for processing graphics (RBG) data
a second pipeline for processing words of video data selectively retrieved from said frame buffer.	circuitry in Parallax 1280/VIPER for processing video (YUV) data

### **CLAIM 43 IS INVALID OVER PARALLAX**

CLAIM 43	PRIOR ART
43. A display controller for interfacing a multi-format frame buffer and a display device, the multi-format frame buffer having on-screen and off-screen areas each for simultaneously storing both graphics and video pixel data, said display controller comprising:	Parallax 1280/VIPER, with frame buffer for simultaneously storing graphics (RGB) and video (YUV), each in on-screen or off-screen areas
circuitry for selectively retrieving pixel data from a selected one of said on-screen and off-screen areas of said frame buffer;	circuitry in Parallax 1280/VIPER for selectively retrieving graphics (RGB) and video (YUV) data, from either on-screen or off-screen regions of the frame buffer
a graphics backend pipeline for processing graphics data retrieved from said selected one of said areas of said frame buffer;	circuitry in Parallax 1280/VIPER for processing graphics (RBG) data

a video backend pipeline for processing video data retrieved from said selected one of said areas of said frame buffer; and	circuitry in Parallax 1280/VIPER for processing video (YUV) data
an output selector for selectively passing to said display device data received from said graphics or video backend pipelines.	circuitry in Parallax 1280/VIPER for selecting between graphics/video pipelines (overlay generator, or multiplexer)

# **CLAIM 1 IS INVALID OVER PARALLAX**

CLAIM 1	PRIOR ART
1. A graphics and video controller comprising:	Parallax 1280/VIPER
an interface for receiving words of pixel data, each said word associated with an address buffer;	circuitry in Parallax 1280/VIPER for receiving pixel data
circuitry for writing each said word of said pixel data received by said interface to a one of on-screen and off-screen memory areas of a frame buffer;	circuitry in Parallax 1280/VIPER for writing pixel data into on-screen and off-screen regions of the frame buffer
circuitry for selectively retrieving said words from said on-screen and off-screen area;	circuitry in Parallax 1280/VIPER for selectively retrieving pixel data from on-screen and off-screen regions of the frame buffer
a first pipeline for processing words of graphics data retrieved from said frame buffer;	circuitry in Parallax 1280/VIPER for processing graphics (RBG) data
a second pipeline for processing words of video data retrieved from said frame buffer.	circuitry in Parallax 1280/VIPER for processing video (YUV) data

### CLAIM 2 IS INVALID OVER PARALLAX

CLAIM 2	PRIOR ART
2. The controller of claim 1 and further comprising output selection circuitry for selecting for output between graphics data received from said first pipeline and data received from said second pipeline, said selection circuitry operable to:	circuitry in Parallax 1280/VIPER for selecting between graphics and video pipelines
in a first mode, pass data from said first pipeline; and	circuitry in Parallax 1280/VIPER for passing data from graphics pipeline
in a second mode, pass data from said second pipeline when said data corresponds to a selected display position of a display window.	circuitry in Parallax 1280/VIPER for passing data from video pipeline when in a video region

### CLAIM 3 IS INVALID OVER PARALLAX

CLAIM 3	PRIOR ART
3. The controller of claim 2 wherein said selection circuitry is further operable to:	

00374041-023939

00374041-032599

in a third mode, pass data from said second pipeline when said data corresponds to said selected display position of said display window and data from said first pipeline match a color key.

b1c

circuitry in Parallax 1280/VIPER for passing data from video pipeline when in a video region and based on graphics over video keying control

In Parallax 1280/VIPER, overlay of video onto graphics with video-based keying was accomplished with Access Attribute control during a copy or frame grabbing operation. This technique differed from the popular "color-key blue" which was well known in the art, while still solving customer needs for keying of video over graphics.

Parallax's choice to key graphics OVER video was a design decision based on customer needs. Nothing unique about this choice over decision to overlay video onto graphics.

# CLAIM 4 IS INVALID OVER PARALLAX

CLAIM 4	PRIOR ART
<p>4. The controller of claim 3 wherein said selection circuitry is further operable in a fourth mode to pass data from said second pipeline when data from said first pipeline match a color key.</p>	<p>circuitry in Parallax 1280/VIPER for passing data from video pipeline based on graphics over video keying control</p> <p>In Parallax 1280/VIPER, overlay of video onto graphics with video-based keying was accomplished with Access Attribute control during a copy or frame grabbing operation. This technique differed from the popular "color-key blue" which was well known in the art, while still solving customer needs for keying of video over graphics.</p> <p>Choice to key graphics OVER video was a design decision based on customer needs. Nothing unique about this choice over decision to overlay video onto graphics.</p>

2025-11-14 14:44:44

### CLAIM 5 IS INVALID OVER PARALLAX

CLAIM 5	PRIOR ART
5. The controller of claim 1 wherein said circuitry for retrieving maintains a stream of graphics data to said first pipeline and provides video data to said second pipeline when a display raster scan reaches said display position of said window.	<p>circuitry in Parallax 1280/VIPER for passing graphics data to graphics pipeline and for passing video data to video pipeline when in a video region</p> <p>Position of video region was indicated by DAM bit in addition to position on raster display scan.</p>

### CLAIM 6 IS INVALID OVER PARALLAX

CLAIM 6	PRIOR ART
6. The controller of claim 1 and further comprising:	
a video port for receiving real-time video data; and	circuitry in Parallax 1280/VIPER for receiving real-time video
circuitry for generating an address to said memory at which said real-time video data is to be stored.	circuitry in Parallax 1280/VIPER for generating addresses for received real-time video data

55329-1404260

### CLAIM 7 IS INVALID OVER PARALLAX

CLAIM 7	PRIOR ART
7. The controller of claim 1 wherein said second pipeline includes a first first-in-first-out memory for receiving data for a first display line of pixels in memory and a second display line of pixels memory.	<p>Parallax hardware employed first-in-first-out buffering between the display memory and the display monitor for the graphics format data and the video format data.</p> <p>Parallax's implementation required only one line of buffering outside the display memory. The addition of a second FIFO buffer would be a requirement of the design implementation, NOT a unique invention.</p>

### CLAIM 8 IS INVALID OVER PARALLAX

CLAIM 8	PRIOR ART
8. The controller of claim 7 wherein said first display line adjacent in memory to said second display line.	<p>Parallax hardware employed first-in-first-out buffering between the display memory and the display monitor for the graphics format data and the video format data.</p> <p>In the event that a given design would need more than a single line of buffering, it would be obvious that the second line of buffering be adjacent to the first (and below the first).</p>

### CLAIM 9 IS INVALID OVER PARALLAX

CLAIM 9	PRIOR ART
9. The controller of claim 7 wherein said output selection circuitry comprises:	

\_\_\_\_\_

an output selector for selecting between data from said second pipeline and data from said first pipeline in response to a selection control signal;	circuitry in Parallax 1280/VIPER for selecting between graphics/video pipelines (overlay generator, or multiplexer)
a register for maintaining a plurality of overlay control bits;	<p>register of plural control bits in Parallax 1280/VIPER</p> <p>There is nothing inventive about the use of a register to hold control bits. Parallax implemented a register specific to the control of the video/graphics MUX.</p>
window position control circuitry for selectively generating a position control signal when a word of said data stream from said second pipeline falls within a display window;	<p>circuitry in Parallax 1280/VIPER for passing data from video pipeline when in a video region</p> <p>Position of video region was indicated by DAM bit in addition to position on raster display scan.</p>
color comparison circuitry for comparing words of said data stream from said first pipeline with a color key and for providing in response a color comparison control signal; and	<p>circuitry in Parallax 1280/VIPER for passing data from video pipeline based on graphics over video keying control</p> <p>In Parallax 1280/VIPER, overlay of video onto graphics with video-based keying was accomplished with Access Attribute control during a copy or frame grabbing operation. This technique differed from the popular "color-key blue" which was well known in the art, while still solving customer needs for keying of video over graphics.</p> <p>Choice to key graphics OVER video was a design decision based on customer needs. Nothing unique about this choice over decision to overlay video onto graphics.</p>



<p>a control selector for selectively providing a said selection control signal in response to said overlay control bits in said register and at least one of said position control and color comparison control signals.</p>	<p>circuitry in Parallax 1280/VIPER for passing data from video pipeline when in a video region and based on graphics-over-video keying control bit.</p> <p>This functionality was selectively provided in response to overlay control bits from the overlay control register.</p> <p>In addition, overlay controls were well known in art, and it was well known in art to store control bits in a register.</p>
---	---

### CLAIM 10 IS INVALID OVER PARALLAX

CLAIM 10	PRIOR ART
10. The controller of claim 9 wherein said window position control circuitry comprises:	
<p>window position counters operable to increment from initial count values corresponding to a starting pixel of a display window as data representing each pixel in a display screen is pipelined through said overlay control circuitry;</p>	<p>window position counters in the Parallax 1280/VIPER for the generation of coherent pixel display streams (which is well known art in graphics controllers).</p> <p>Parallax 1280/VIPER hardware was able to maintain a plurality of video display start coordinates within a single display, each with arbitrary X and Y origins.</p>
<p>screen position counters operable to count as data representing each pixel in said display screen is pipelined through said overlay control circuitry; and</p>	<p>window position counters in the Parallax 1280/VIPER for the generation of coherent pixel display streams (which is well known art in graphics controllers).</p> <p>Parallax 1280/VIPER hardware was able to maintain a plurality of video display start coordinates within a single display, each with arbitrary X and Y origins</p>

comparison circuitry operable to compare a current count in said window position counters and a current count in said screen position counters and selectively generate said position control signal in response.

circuitry in Parallax 1280/VIPER for loading counters with difference, and comparing result to zero

This form of compare circuit is well known in the art.

### CLAIM 12 IS INVALID OVER PARALLAX

CLAIM 12	PRIOR ART
12. The controller of claim 1 wherein said interface includes a dual-aperture port.	circuitry in Parallax 1280/VIPER for receiving graphics and video data through multi-aperture port

### CLAIM 13 IS INVALID OVER PARALLAX

13. A controller comprising:	
circuitry for writing selectively each word of received data into [a] selected one of on-screen and off-screen memory spaces of a frame buffer:	circuitry in Parallax 1280/VIPER qualifying each write to on-screen and off-screen areas of display memory through the Access Attribute Control
a first port for receiving video and graphics data, a word of said data received with an address of said memory spaces directing said word to be processed as a word of video data or a word of graphics data;	circuitry in Parallax 1280/VIPER for receiving graphics and video data and directing that it be stored and displayed as video or graphics
a second port for receiving real-time video data:	circuitry in Parallax 1280/VIPER for receiving real-time video data
circuitry for generating an address associated with a selected one of said memory spaces for a work of said real-time video data;	circuitry in Parallax 1280/VIPER for generating addresses for received real-time video data

00374044-032B07

<p>circuitry for selectively retrieving said words of data from said on-screen and off-screen memory spaces as data is rastered for driving a display;</p>	<p>circuitry in Parallax 1280/VIPER for selectively retrieving graphics (RGB) and video (YUV) data, from either on-screen or off-screen regions of the frame buffer</p>
<p>a graphics backend pipeline for processing ones of said words of data representing graphics data retrieved from said frame buffer;</p>	<p>circuitry in Parallax 1280/VIPER for processing graphics (RBG) data</p>
<p>a video backend pipeline for processing other ones of said words of data representing video data retrieved from said frame buffer, said circuitry for retrieving always rastering a stream of data from said frame buffer to said graphics backend pipeline and rastering video data to said video backend pipeline when a display raster scan reaches a display position of a window; and</p>	<p>circuitry in Parallax 1280/VIPER for processing video (YUV) data, for passing graphics data to graphics pipeline, and for passing data to video pipeline when in a video region.</p> <p>Position of video region was indicated by DAM bit in addition to position on raster display scan.</p>
<p>output selector circuitry for selecting for output between words of data output from said graphics backend pipeline and words of data output from said video backend pipeline.</p>	<p>circuitry in Parallax 1280/VIPER for selecting between graphics/video pipelines (overlay generator, or multiplexer)</p>

---

27

# CLAIM 15 IS INVALID OVER PARALLAX

037404-0369

CLAIM 15	PRIOR ART
15. The controller of claim 13 wherein said output selector is operable to:	
in a first mode pass only a word of data output from said graphics pipeline;	circuitry in Parallax 1280/VIPER for passing data from graphics pipeline
in a second mode pass a word of data output from said video pipeline when said display raster scan has reached a display position corresponding to a window and a word of data from said graphics pipeline when said display raster scan is in any other display position;	circuitry in Parallax 1280/VIPER for passing data from video pipeline when in a video region and from graphics pipeline elsewhere
in a third mode pass a word of data output from said video pipeline when said display raster scan has reached a display position corresponding to a window and a corresponding word of data from said graphics pipeline matches a color key and a word of data from said graphics pipeline when said display raster scan is in any other display position; and	<p>circuitry in Parallax 1280/VIPER for passing data from video pipeline when in a video region and based on graphics over video keying control and from graphics pipeline elsewhere</p> <p>In Parallax 1280/VIPER, overlay of video onto graphics with video-based keying was accomplished with Access Attribute control during a copy or frame grabbing operation. This technique differed from the popular "color-key blue" which was well known in the art, while still solving customer needs for keying of video over graphics.</p> <p>Choice to key graphics OVER video was a design decision based on customer needs. Nothing unique about this choice over decision to overlay video onto graphics.</p>

<p>in a fourth mode pass a word of data from said video pipeline when said corresponding word of data from said graphics pipeline matches a color key and a word of data from said graphics pipeline when said display raster scan is in any other display position.</p>	<p>circuitry in Parallax 1280/VIPER for passing data from video based on graphics over video keying control and from graphics pipeline elsewhere</p> <p>In Parallax 1280/VIPER, overlay of video onto graphics with video-based keying was accomplished with Access Attribute control during a copy or frame grabbing operation. This technique differed from the popular "color-key blue" which was well known in the art, while still solving customer needs for keying of video over graphics.</p> <p>Choice to key graphics OVER video was a design decision based on customer needs. Nothing unique about this choice over decision to overlay video onto graphics.</p>
--	--

### CLAIM 16 IS INVALID OVER PARALLAX

CLAIM 16	PRIOR ART
<p>16. The controller of claim 13 wherein said video pipeline includes a first first-in-first-out memory for receiving a plurality of words of data for a first display line of pixels in memory and a second first-in-first-out memory or receiving a plurality of words of data from a second display line of pixels in memory.</p>	<p>Parallax 1280/VIPER hardware employed first-in-first-out buffering between the display memory and the display monitor for the graphics format data and the video format data.</p> <p>Parallax's implementation required only one line of buffering outside the display memory. The addition of a second FIFO buffer would be a requirement of the design implementation, NOT a unique invention.</p>

### CLAIM 17 IS INVALID OVER PARALLAX

CLAIM 17	PRIOR ART
17. The controller of claim 16 wherein said first display line is stored adjacent in memory to said second display line.	<p>Parallax 1280/VIPER hardware employed first-in-first-out buffering between the display memory and the display monitor for the graphics format data and the video format data.</p> <p>In the event that a given design would need more than a single line of buffering, it would be obvious that the second line of buffering be adjacent to the first (and below the first).</p>

### CLAIM 18 IS INVALID OVER PARALLAX

CLAIM 18	PRIOR ART
18. The controller of claim 13 wherein said output selector circuitry comprises:	
a control selector having a plurality of control inputs coupled to a register said register storing a plurality of overlay control bits:	<p>circuitry in Parallax 1280/VIPER for passing data from video pipeline when in a video region and based on graphics-over-video keying control bit.</p> <p>This functionality was selectively provided in response to overlay control bits from the overlay control register.</p> <p>In addition, overlay controls were well known in art, and it was well known in art to store control bits in a register.</p>

<p>window position control circuitry coupled to a first control input of said control selector; said window position control circuitry operable to selectively provide a first control signal to said first control input when a word of data being pipelined through said video pipeline falls within a display window;</p>	<p>circuitry in Parallax 1280/VIPER for passing data from video pipeline when in a video region</p> <p>Position of video region was indicated by DAM bit in addition to position on raster display scan.</p>
<p>color comparison circuitry operable to compare a word of data being pipelined through said graphics pipeline with a color key and provide in response a second control signal to a second control input of said control selector; and</p>	<p>circuitry in Parallax 1280/VIPER for passing data from video pipeline based on graphics over video keying control</p> <p>In Parallax 1280/VIPER, overlay of video onto graphics with video-based keying was accomplished with Access Attribute control during a copy or frame grabbing operation. This technique differed from the popular "color-key blue" which was well known in the art, while still solving customer needs for keying of video over graphics.</p> <p>Choice to key graphics OVER video was a design decision based on customer needs. Nothing unique about this choice over decision to overlay video onto graphics.</p>
<p>wherein said control selector is operable to provide an output selection control signal in response to at least one of said first and second control signals and said overlay control bits being stored in said register.</p>	<p>circuitry in Parallax 1280/VIPER for selecting between graphics/video pipelines (overlay generator, or multiplexer)</p> <p>This functionality was selectively provided in response to overlay control bits from the overlay control register.</p> <p>In addition, overlay controls were well known in art, and it was well known in art to store control bits in a register.</p>

### CLAIM 19 IS INVALID OVER PARALLAX

CLAIM 19	PRIOR ART
19. The circuitry of claim 18 wherein said output selector circuitry further includes at third control input coupled to certain bits of said graphics pipeline, said output selector further operable to select between data on said respective video and graphics pipelines in response to said certain bits presented to said selector circuitry.	<p>circuitry in Parallax 1280/VIPER for selecting between graphics/video pipelines (overlay generator, or multiplexer)</p> <p>In the event that a given design would need more than two control variables, it would be obvious to use a third control signal.</p>

### CLAIM 20 IS INVALID OVER PARALLAX

CLAIM 20	PRIOR ART
20. The circuitry of claim 18 wherein said window position control circuitry comprises:	
a window x-position counter operable to count from a loaded x-position value in response to a video clock, said x-position counter reloading in response to display horizontal synchronization signal;	<p>window position counters in the Parallax 1280/VIPER for the generation of coherent pixel display streams (which is well known art in graphics controllers).</p> <p>Parallax 1280/VIPER hardware was able to maintain a plurality of video display start coordinates within a single display, each with arbitrary X and Y origins.</p>
a window y-position counter operable to count from a loaded y-position value in response to said horizontal synchronization signal, said y-position counter reloading in response to a display vertical synchronization signal;	<p>window position counters in the Parallax 1280/VIPER for the generation of coherent pixel display streams (which is well known art in graphics controllers).</p> <p>Parallax 1280/VIPER hardware was able to maintain a plurality of video display start coordinates within a single display, each with arbitrary X and Y origins.</p>

09374044.083599



CRT position circuitry operable maintain counts corresponding to a current display pixel; and	<p>window position counters in the Parallax 1280/VIPER for the generation of coherent pixel display streams (which is well known art in graphics controllers).</p> <p>Parallax 1280/VIPER hardware was able to maintain a plurality of video display start coordinates within a single display, each with arbitrary X and Y origins.</p>
comparison circuitry operable to compare current counts in said window counters with said current counts held in said CRT position circuitry and generate in response said first control signal.	<p>circuitry in Parallax 1280/VIPER for loading counters with difference, and comparing result to zero</p> <p>This form of compare circuit is well known in the art.</p>

### CLAIM 21 IS INVALID OVER PARALLAX

09374041-082639  
F69780-1404/2560

CLAIM 21	PRIOR ART
21. The circuitry of claim 20 wherein said window position control circuitry further comprises an x-position register for holding said x-position value for loading into said x-position counter and a y-position register for holding said y-position value for loading into said y-position counter.	<p>window position counters in the Parallax 1280/VIPER for the generation of coherent pixel display streams (which is well known art in graphics controllers).</p> <p>Placing registers to store load values for counters is not inventive, and had been implemented many times by Parallax's hardware design engineers.</p> <p>Parallax 1280/VIPER hardware was able to maintain a plurality of video display start coordinates within a single display, each with arbitrary X and Y origins.</p>

# CLAIM 23 IS INVALID OVER PARALLAX

CLAIM 23	PRIOR ART
23. The circuitry of claim 13 wherein said video pipeline comprises:	
a first-in/first-out memory for receiving a first stream of words of data from said frame buffer;	Parallax 1280/VIPER hardware employed first-in-first-out buffering between the display memory and the display monitor for the graphics format data and the video format data.
a second first-in/first-out memory disposed in parallel with said first first-in/first-out memory for receiving a second stream of words of data from said frame buffer; and	<p>Parallax 1280/VIPER hardware employed first-in-first-out buffering between the display memory and the display monitor for the graphics format data and the video format data.</p> <p>Parallax's implementation required only one line of buffering outside the display memory. The addition of a second FIFO buffer would be a requirement of the design implementation, NOT a unique invention.</p>
interpolation circuitry for selectively generating an additional word of data by interpolating a word of said first stream and a word of second stream data output from said first and second first-in/first-out memories.	Parallax's implementation did not interpolate display lines between sampled lines. Parallax's implementation used direct samples from the line above or the line below. However, use of interpolation filters for spatially expanding a display area was well known art as of September 1993, and can be found in numerous video special effect generators (for example Abekas 8150).

## CLAIM 24 IS INVALID OVER PARALLAX

CLAIM 24	PRIOR ART
24. The controller of claim 13 wherein said first port comprises a dual-aperture port.	circuitry in Parallax 1280/VIPER for receiving graphics and video data through multi-aperture port

I understand that I may be asked to prepare a rebuttal report and/or give rebuttal testimony at a hearing on matters not covered in this expert report. Additionally, I understand that discovery has not been completed and that I will consider additional evidence in connection with the issues discussed above. After considering any such additional evidence, I may supplement this report as necessary.

Respectfully submitted,

Date: November 17, 1998

  
William G. Mears

99748 P11



Pixel  
Semiconductor  
A Cirrus Logic Company

# CL-PX2070

Preliminary Data Book

## APPLICATIONS

- Presentation
- Video Editing
- Video Authoring
- Video Teleconferencing
- Interactive Education Systems
- Games

## FEATURES

- Extensive software support available — contact Cirrus Logic Sales office for complete details
- Supports up to three simultaneous video data streams
- Video scaling
- Supports both YCbCr and RGB formats
- Interfaces to CODECs, decoders, encoders
- Integrated ISA, MCA, and host bus interfaces

(cont. next page)

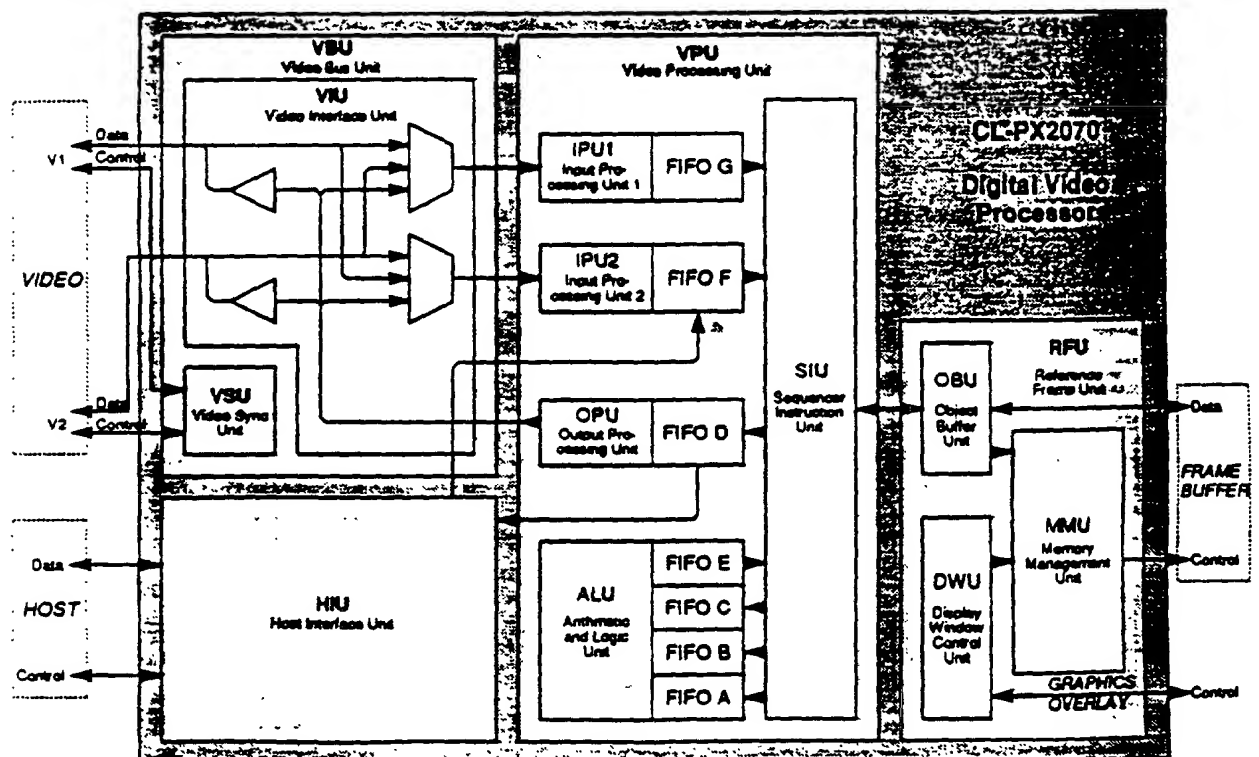
## Digital Video Processor

## OVERVIEW

The CL-PX2070 Digital Video Processor (DVP) provides a powerful, cost-effective desktop solution for computer graphics and imaging. The DVP can be used in presentations, video teleconferencing, animation, and video capture for scaling with video signal processors dedicated to compressing and decompressing video data streams.

(cont. next page)

Functional Block Diagram





## FEATURES (cont.)

- Complete frame buffer control
- 1/2 - 8 Mbytes of frame buffer memory
- Video stream format conversion
- Color space conversion
- Supports up to eight simultaneous object buffers
- Programmable, triple-channel LUT RAM
- Prescaling, zoom, and windowing
- Graphic and bitmapped stream support
- Programmable sync slave or master
- When used with the CL-PX2080 MediaDAC™
  - Simultaneous video and graphics display
  - Four simultaneous, overlapping (occluded) display windows
  - Zooms from 1x to 256x
  - 1024 x 768 display at 85 MHz

## OVERVIEW (cont.)

The DVP combines the real-time video scaling features of the CL-PX0072 VWG with the frame buffer memory management, arithmetic and logical processing, a programmable host system bus interface, flexible mainstream video datapath, and windowing control for multiple, simultaneous video data streams.

The DVP has four major functional units:

- HIU: Host Interface Unit
- VBU: Video Bus Unit
- VPU: Video Processing Unit
- RFU: Reference Frame Unit

### HIU: Host Interface Unit

The HIU interfaces the DVP to the host system. It transfers graphic or video data between the host system and the frame buffer through direct access to FIFOs in the VPU, and accesses the DVP control registers.

### VBU: Video Bus Unit

The VBU manages the flow of video and graphic streams between the DVP and up to three independent devices (including the host system).

The VBU provides two independent, real-time video I/O ports (V1 and V2), and contains two sub-units — the VIU and VSU.

V1 and V2 have the following characteristics:

- Each can be configured as input only, output only, or pixel- or field-duplexed I/O;
- Each provides programmable sync polarity;
- Either port can use the VSU sync generator;
- Each supports the following video formats:
  - *Input:* YCbCr 16-bit 4:2:2, 12-bit 4:1:1; RGB 16-bit, 8-bit;
  - *Output:* YCbCr 16-bit; RGB 16-bit, 8-bit;
- V2 controls the video stream data flow between the DVP and typical CODEC devices.

	ISA Bus	MCA Bus	Local Hardware
Interface	DVP interfaces with the host system interface bus.		DVP interfaces with the processor bus.
Multiplex Support	DVP signals support the required host system address/data impleading, and provide bidirectional buffering of the host system data bus.		N/A
Address Decode	DVP internally decodes the bus address during system I/O cycles.		The host system provides the decoded chip select signal for use with register select input signals.



*The VIU (Video Interface Unit) controls the flow of internal video streams through the video ports to all external devices. It controls:*

- the source and direction of video stream and sync control inputs;
- the field-toggling mode and field ID signals;
- the watchdog timer feature.

Two VIU master control registers provide matching fields that specify input and output sync modes.

*The VSU (Video Sync Unit) implements identical, independent reference signals for each video port:*

- Vertical sync signals specify the beginning of a field or frame.
- Horizontal sync signals specify the beginning of a line.
- Horizontal/composite blanking signals specify the horizontal/composite blanking interval.

#### **VPU: Video Processing Unit**

The VPU processes field-oriented video. It can simultaneously process two external, bidirectional real-time video streams and a single external, bidirectional host video or graphic data stream. It also provides a data path between the DVP and the host system for bidirectional graphic streams through the HIU. FIFO D can send to, and FIFO F can receive from the HIU directly.

The VPU has five subunits — the IPU1, IPU2, OPU, ALU, and SIU.

*The IPU1 (Input Processor Unit 1) prepares an input video stream for ALU processing and/or storage in the frame buffer, then outputs the prepared stream to the frame buffer data bus. Its video processing features include:*

- YCbCr and RGB input stream format conversion,
- color space conversion,
- programmable data tagging,
- three-channel lookup table operations,
- horizontal prescaling,
- window clipping,
- horizontal and vertical scaling, and
- output stream format conversion.

*The IPU2 (Input Processor Unit 2) controls prescaling and windowing.*

*The OPU (Output Processing Unit) controls zoom, window clipping, and output format functions.*

*The ALU (Arithmetic Logic Unit) performs arithmetic, logical, and tagging operations for YCbCr streams, and logical and tagging operations only for RGB and 8-bit pseudocolor streams. It controls stream format, operand source selection, tagging operation selection, and arithmetic or logical operation for both field times, and can process up to three simultaneous video streams input through its FIFOs.*

*The SIU (Sequencer Instruction Unit) is a special-purpose microcontroller that coordinates the flow of multiple, simultaneous data streams between the IPU1, IPU2, OPU, ALU, and OBU.*

The SIU is field-based when processing interlaced video data; that is, it distinguishes between the vertical sync pulses for each field and executes one of two different instruction sequences, causing multiple stream flows to appear concurrent.

#### **RFU: Reference Frame Unit**

The RFU provides simultaneous access to eight object buffers and four display windows. It has three subunits — the OBU, DWU, and MMU.

*The OBU (Object Buffer Unit) specifies the size, location, operating mode, X and Y raster directions, FIFO association, chrominance and luminance channel masking, and output decimation for each object buffer. It allows each object buffer to be locked to either video source, or to be programmed to operate independently. Object buffers can also be placed anywhere within the linearly-addressable frame buffer.*

*The DWU (Display Window Unit) allows each display window to be any size or location. These display windows can overlap when the DVP is used with the CL-PX2080 MediaDAC™.*

*The MMU (Memory Management Unit) provides the frame buffer control interface for up to 8 megabytes of DRAM or VRAM.*

ATI019029



## TABLE OF CONTENTS

LIST OF FIGURES .....	7
LIST OF TABLES .....	7
CONVENTIONS, ABBREVIATIONS, AND TRADEMARKS .....	8
1. PIN INFORMATION.....	10
1.1 Pin Diagram .....	10
1.2 DVP Functional Signal Groups .....	11
1.3 Pin Assignment Table .....	11
2. DETAILED SIGNAL DESCRIPTIONS .....	14
2.1 Host Interface — ISA .....	14
2.2 Host Interface — MCA .....	15
2.3 Host Interface — Local Hardware .....	16
2.4 Graphics Overlay Interface .....	17
2.5 Video Interface .....	18
2.6 Frame Buffer Interface .....	19
2.7 Power and Ground .....	19
3. FUNCTIONAL DESCRIPTION .....	20
4. DETAILED REGISTER DESCRIPTIONS.....	22
4.1 HIU: Host Interface Unit .....	22
4.1.1 HIU_CSU: Configuration Setup.....	23
4.1.2 HIU_DBG: Debug Control .....	24
4.1.3 HIU_DRD: Debug Read .....	24
4.1.4 HIU_IRQ: Interrupt Request .....	25
4.1.5 HIU_OCS: Operation Control/Status.....	26
4.1.6 HIU_RIN: Register Index .....	27
4.1.7 HIU_RDT: Register Data Port.....	28
4.1.8 HIU_MDT: Memory Data Port .....	28
4.1.9 HIU_ISU: Interrupt Setup .....	29
4.2.1 VIU: Video Interface Unit.....	30
4.2.1.1 VIU_MCRp: VIU Master Control .....	30
4.2 VBU: Video Bus Unit .....	30
4.2.1.2 VIU_DPCf: Datapath Control .....	32
4.2.1.3 VIU_WDT: Watchdog Timer .....	33
4.2.1.4 VIU_TEST: Test Register .....	34
4.2.2 VSU: Video Sync Unit .....	35
4.2.2.1 VSU_HSW: Horizontal Sync Width .....	35
4.2.2.2 VSU_HAD: Horizontal Active Delay .....	36
4.2.2.3 VSU_HAP: Horizontal Active Pixels .....	36
4.2.2.4 VSU_HP: Horizontal Period .....	36
4.2.2.5 VSU_VSW: Vertical Sync Width .....	37
4.2.2.6 VSU_VAD: Vertical Active Delay .....	37
4.2.2.7 VSU_VAP: Vertical Active Pixels .....	38
4.2.2.8 VSU_VP: Vertical Period .....	38
4.3 VPU: Video Processor Unit.....	39
4.3.1 VPU Global Control.....	44
4.3.1.1 VPU_MCR: VPU Master Control.....	44
	ATI019030



4.3.2	IPU1: Input Processor Unit .....	45
4.3.2.1	IPU1_PIX: Pixel Count .....	45
4.3.2.2	IPU1_LIC: Line Count .....	45
4.3.2.3	IPU1_FLC: Field Count .....	46
4.3.2.4	IPU1_LIR: Line Count Interrupt Request .....	46
4.3.2.5	IPU1_FIR: Field Count Interrupt Request .....	48
4.3.2.6	IPU1_LRB: LUT RAM Base Address .....	47
4.3.2.7	IPU1_LRD: LUT RAM Data .....	47
4.3.2.8	IPU1_MCRf: IPU1 Master Control .....	48
4.3.2.9	IPU1_XBnf: X Begin .....	49
4.3.2.10	IPU1_XEIf: X End .....	50
4.3.2.11	IPU1_XSnf: X Shrink .....	50
4.3.2.12	IPU1_YBnf: Y Begin .....	51
4.3.2.13	IPU1_YEIf: Y End .....	51
4.3.2.14	IPU1_YSnf: Y Shrink .....	52
4.3.2.15	IPU1_KFCf: Key Function Code .....	52
4.3.2.16	IPU1_MMxf: Chroma Key Max/Min .....	53
4.3.3	IPU2: Input Processing Unit .....	54
4.3.3.1	IPU2_PIX: Pixel Count .....	54
4.3.3.2	IPU2_LIC: Line Count .....	54
4.3.3.3	IPU2_FLC: Field Count .....	54
4.3.3.4	IPU2_LIR: Line Count Interrupt Request .....	55
4.3.3.5	IPU2_FIR: Field Count Interrupt Request .....	55
4.3.3.6	IPU2_MCRf: IPU2 Master Control .....	56
4.3.3.7	IPU2_XBIf: X Begin .....	56
4.3.3.8	IPU2_XEIf: X End .....	57
4.3.3.9	IPU2_YBIf: Y Begin .....	57
4.3.3.10	IPU2_YEIf: Y End .....	58
4.3.4	SIU: Sequencer Instruction Unit .....	58
4.3.4.1	SIU_MCR: SIU Master Control .....	58
4.3.4.2	SIU_FCS: FIFO Control/Status .....	59
4.3.4.3	SIU_FOU: FIFO Overflow/Underflow .....	60
4.3.4.4	SIUs_SIM: Sequencer Instruction Memory .....	81
4.3.4.5	SIU_FAR: FIFO Auto Reset .....	62
4.3.5	ALU: Arithmetic and Logic Unit .....	62
4.3.5.1	ALU_MCRf: ALU Master Control .....	62
4.3.5.2	ALU_TOP: Tag Operation .....	64
4.3.5.3	ALU_AV: Alpha Value .....	64
4.3.5.4	ALU_LOPx: Logic Operation .....	65
4.3.5.5	ALU_CAx: Constant A .....	65
4.3.5.6	ALU_CBx: Constant B .....	66
4.3.5.7	ALU_CCx: Constant C .....	66
4.3.6	OPU: Output Processing Unit .....	67
4.3.6.1	OPU_MCRf: OPU Master Control .....	67
4.3.6.2	OPU_XBIf: X Begin .....	68
4.3.6.3	OPU_XEIf: X End .....	68
4.3.6.4	OPU_YBIf: Y Begin .....	69
4.3.6.5	OPU_YEIf: Y End .....	69
4.3.7	RFU: Reference Frame Unit .....	70

ATI019031





4.3.8	MMU: Memory Management Unit .....	73
4.3.8.1	MMU_MCR: MMU Master Control .....	73
4.3.9	OBU: Object Buffer Unit .....	74
4.3.9.1	OBUo_MCR: Object Buffer Master Control .....	74
4.3.9.2	OBUo_RFX: Object Buffer Reference Frame Size .....	75
4.3.10	OBUo_LsB: Object Buffer Linear Start Address .....	76
4.3.10.1	OBUo_BsA: Object Buffer Size .....	77
4.3.10.2	OBUo_DEC: Object Buffer Decimate Control .....	78
4.3.11	DWU: Display Window Unit .....	78
4.3.11.1	DWU_MCR: Display Window Master Control .....	78
4.3.11.2	DWU_HCR: Display Window Horizontal Control .....	80
4.3.11.3	DWUd_DZF: Display Window Display Zoom Factor .....	81
4.3.11.4	DWUd_RFX: Display Window Reference Frame Size .....	81
4.3.11.5	DWUd_LsB: Display Window Linear Start Address .....	82
4.3.11.6	DWUd_WsA: Display Window Size .....	82
4.3.11.7	DWUd_DsA: Display Window Start .....	83
5	<b>ELECTRICAL SPECIFICATIONS .....</b>	<b>84</b>
5.1	Absolute Maximum Ratings .....	84
5.2	DVP Specifications (Digital) .....	84
5.3	AC Characteristics/Timing Information .....	85
5.3.1	Index of Timing Information .....	85
5.3.2	ISA Bus Timing .....	86
5.3.3	MCA Bus Timing .....	88
5.3.4	Local Hardware Interface Timing .....	90
5.3.5	Video Port Timing .....	92
5.3.6	Memory Timing .....	94
6	<b>PACKAGE DIMENSIONS — 160-Lead PQFP .....</b>	<b>98</b>
7	<b>ORDERING INFORMATION .....</b>	<b>99</b>
A	<b>APPENDIX A. DVP REGISTERS — QUICK REFERENCE .....</b>	<b>100</b>
B	<b>INDEX .....</b>	<b>102</b>
C	<b>INDEX OF CONTROL REGISTERS .....</b>	<b>109</b>
D	<b>INDEX OF REGISTER FIELDS .....</b>	<b>111</b>

ATI019032



## LIST OF FIGURES

Figure 1-1.	DVP Pin Diagram .....	10
Figure 4-1.	VSU Horizontal Sync Timing .....	35
Figure 4-2.	VSU Vertical Sync Timing .....	35
Figure 5-1.	ISA Bus — I/O Timing .....	87
Figure 5-2.	MCA Bus — I/O Timing .....	89
Figure 5-3.	MCA Bus — CDSFDBK* Timing.....	89
Figure 5-4.	MCA Bus — CDSETUP* Timing.....	89
Figure 5-5.	Local Hardware Interface — Write Timing.....	90
Figure 5-6.	Local Hardware Interface — Read Timing.....	91
Figure 5-7.	Video I/O Timing.....	93
Figure 5-8.	STALL* and STALLRQ* Timing.....	93
Figure 5-9.	Video and Graphics Port Timing.....	93
Figure 5-10.	Read Transfer Cycle Timing .....	94
Figure 5-11.	CAS* Before RAS* Refresh Timing .....	95
Figure 5-12.	Memory Read Timing .....	97
Figure 5-13.	Memory Write Timing.....	97
Figure 6-1.	DVP Package Information .....	98
Figure 6-2.	DVP Package Information (Expanded View) .....	99

## LIST OF TABLES

Table 4-1.	HIU Register Address Map .....	22
Table 4-2.	HIU Registers Accessed by the Register Data Port .....	22
Table 5-1.	ISA Bus Timing .....	86
Table 5-2.	MCA Bus Timing .....	88
Table 5-3.	Local Hardware Interface — Write Timing .....	90
Table 5-4.	Local Hardware Interface — Read Timing .....	91
Table 5-5.	Video Port Timing .....	92
Table 5-6.	Read Transfer Cycle Timing .....	94
Table 5-7.	CAS* Before RAS* Refresh Timing .....	95
Table 5-8.	Memory Read and Write Timing .....	96

ATI019033



## CONVENTIONS, ABBREVIATIONS, AND TRADEMARKS

### CONVENTIONS

**VIU\_DPCf** Register names containing lower case variables represent groups of registers with similar functions. For example, VIU\_DPCf represents both registers — VIU\_DPC1 (Datapath Control, Field 1) and VIU\_DPC2 (Datapath Control, Field 2). In this data book, the following register variables are used:

a	(axis)	-	X, Y
b	(byte)	-	L (Low) or H (High)
c	(color space)	-	Y, U, V or R, G, B
d	(display window)	-	0:3
f	(field)	-	1:2
n	(number)	-	F (Fraction) or I (Integer)
o	(object buffer)	-	0:7
p	(port)	-	1:2
s	(SIM)	-	0:31
x	(channel)	-	Y, U, V

### ABBREVIATIONS, ACRONYMS, and MNEMONICS

ALU	Arithmetic and Logic Unit
CODEC	COde/DEcode or Compress/decompress
CPU	Central Processing Unit
CRT	Cathode Ray Tube
CTAG	Control TAG multiplexer signal
DRAM	Dynamic Random Access Memory
DWU	Display Window Unit
FBD	Frame Buffer Data
FIFO	First In, First Out
ISA	Industry Standard Architecture
I/O	Input/Output
LSA	Linear Start Address
JPEG	Joint Photographic Expert Group

ATI019034

**CL-PX2070**  
*Digital Video Processor*



**Pixel  
Semiconductor**  
*A Cirrus Logic Company*

LSB	Least Significant Byte
LSb	Least Significant bit
LUT	Look-Up Table
MCA	Micro Channel Architecture
MMU	Memory Management Unit
MSB	Most Significant Byte
MSb	Most Significant bit
OPU	Output Processor Unit
OTAG	Output TAG multiplexer signal
IPU1	Input Processor Unit 1
IPU2	Input Processor Unit 2
POS	Programmable Option Select
PQFP	Plastic Quad Flat Pack
PSE	PreScaler Enable
RGB	Red, Green, Blue
RAM	Random Access Memory
RFU	Reference Frame Unit
SIM	Sequencer Instruction Memory
SIU	Sequencer Instruction Unit
VPU	Video Processor Unit
VRAM	Video dynamic Random Access Memory
YCbCr	Components of the CCIR601 color representation standard. Y = luminance; CbCr = chrominance Y-blue, chrominance Y-red

## TRADEMARKS

MediaDAC™ is a trademark of Pixel Semiconductor, Inc.

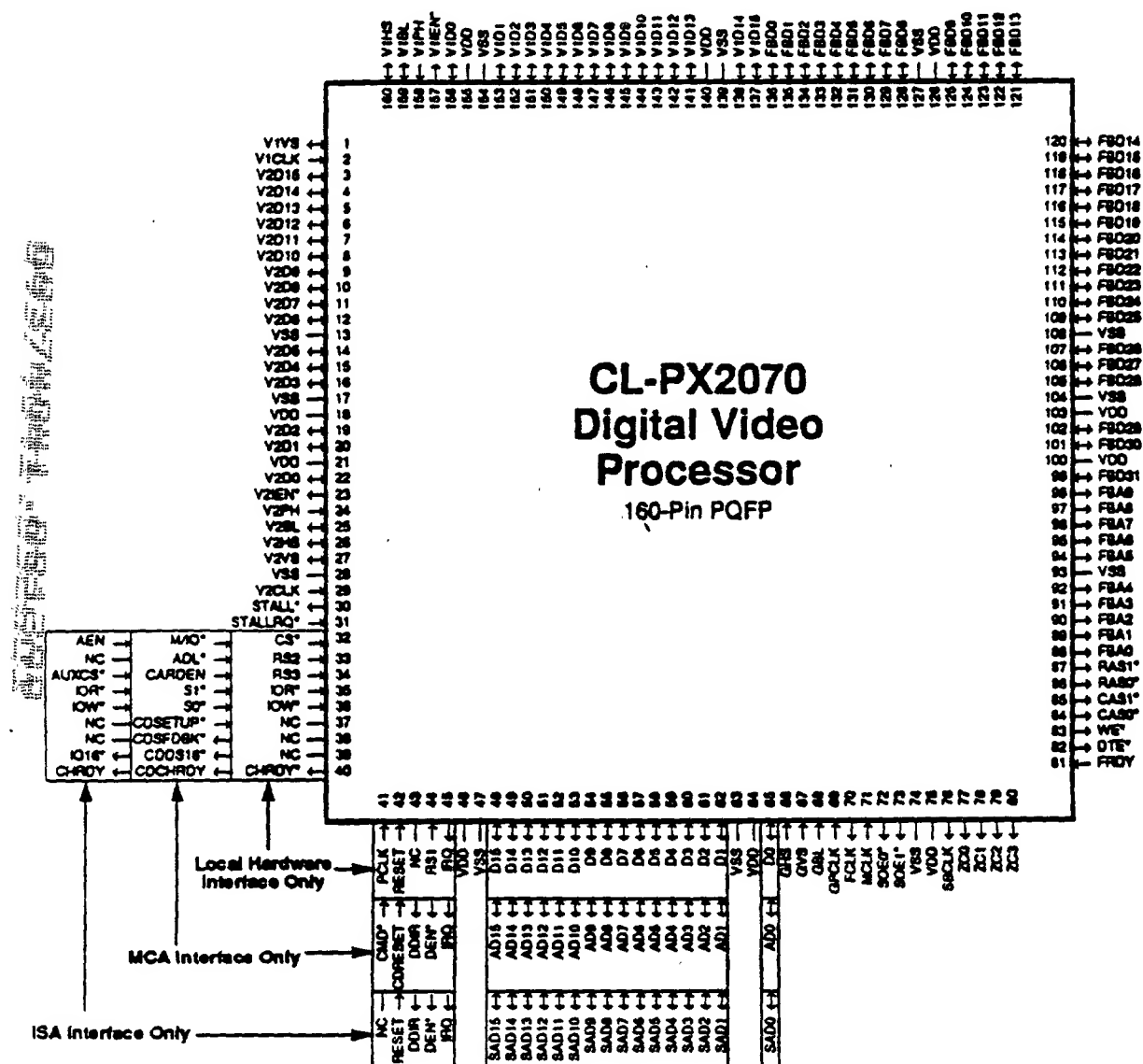
ATI019035



The CL-PX2070 DVP is available in a 160-lead Plastic Quad Flat Pack (PQFP) surface-mount package. It can be configured for ISA, MCA, and local hardware configurations, as shown in Figure 1-1.

**NOTE:** (\*) denotes active-low signals.

## 1.1 Pin Diagram



### Figure 1-1. DVP Pin Diagram

ATI019036





The following conventions are used in the pin assignment table:

- (\*) = active-low signal
- I = input
- O = output
- PWR = power
- TTL = the pad has standard TTL input threshold and output levels
- OD = open drain, TTL inputs
- 4 = 4-mA sink and 2-mA source drive capability
- 24 = 24-mA sink and 8-mA source drive capability

NAME			PIN	TYPE	CELL	FUNCTION
<b>HOST INTERFACE</b>						
<b>ISA</b>	<b>MCA</b>	<b>LOCAL</b>				
<i>Address/Data</i>						
SAD[15:0]	AD[15:0]	—	48:62, 65	I/O	TTL, 4	Address/Data Bus
—	—	D[15:0]	48:62, 65	I/O	TTL, 4	Data Bus
<i>Control</i>						
DEN*	DEN*	—	44	OD	TTL, 8	Data Buffer Enable
—	—	RS1	44	I	TTL	Register Select
DDIR	DDIR	—	43	OD	TTL, 8	Data Buffer Direction
—	—	NC	43	N/A	N/A	No Connect (must be left floating)
IOR*	—	IOR*	35	I	TTL	I/O Read
—	S1*	—	35	I	TTL	Status 1
IOW*	—	IOW*	36	I	TTL	I/O Write
—	S0*	—	36	I	TTL	Status 0
AEN	—	—	32	I	TTL	Address Enable
—	MIO*	—	32	I	TTL	Memory or I/O Cycle
—	—	CS*	32	I	TTL	Chip Select
NC	—	NC	38	N/A	N/A	No Connect (must be left floating)
—	CDSFDBK*	—	38	O	TTL, 4	Card Select Feedback
NC	—	NC	37	N/A	N/A	No Connect (must be left floating)
—	CDSETUP*	—	37	I	TTL	Card Setup
IRQ	IRQ	IRQ	45	O	TTL, 4	Interrupt Request
CHRDY	CDCHRDY	CHRDY*	40	OD	TTL, 24	Channel Ready
IO16*	—	—	39	OD	TTL, 24	16-bit I/O Cycle
—	CDD16*	—	39	OD	TTL, 24	Card Data Size
—	—	NC	39	N/A	N/A	No Connect (must be left floating)
NC	—	—	41	N/A	N/A	No Connect (must be left floating)
—	CMD*	—	41	I	TTL	Command
—	—	PCLK	41	I	TTL	Processor Clock
RESET	CDRESET	RESET	42	I	TTL	Reset
NC	—	—	33	N/A	N/A	No Connect (must be left floating)
—	ADL*	—	33	I	TTL	Address Latch
—	—	RS2	33	I	TTL	Register Select
AUXCS*	—	—	34	I	TTL	Auxiliary Chip Select
—	CARDEN	—	34	I	TTL	Card Enable
—	—	RS3	34	I	TTL	Register Select

ATI019038

**CL-PX2070**  
Digital Video Processor



**Pixel Semiconductor**  
A Cirrus Logic Company

NAME	PIN	TYPE	CELL	FUNCTION
<b>GRAPHICS OVERLAY INTERFACE</b>				
GPCLK	69	I	TTL	Graphics Pixel Clock
GVS	67	I	TTL	Graphics Vertical Sync
GHS	66	I	TTL	Graphics Horizontal Sync
GBL	68	I	TTL	Graphics Blanking
ZC[3:0]	80:77	O	TTL, 4	Zoom Control Bus
<b>VIDEO INTERFACE</b>				
<i>Data</i>				
V1D[15:0]	137:138, 141:153, 156	I/O	TTL, 4	V1 (Video Port 1) Data Bus
V2D[15:0]	3:12, 14:16, 19:20, 22	I/O	TTL, 4	V2 (Video Port 2) Data Bus
<i>Control</i>				
V1VS	1	I/O	TTL, 4	V1 Vertical Sync
V2VS	27	I/O	TTL, 4	V2 Vertical Sync
V1HS	160	I/O	TTL, 4	V1 Horizontal Sync
V2HS	26	I/O	TTL, 4	V2 Horizontal Sync
V1BL	159	I/O	TTL, 4	V1 Horizontal/Composite Blanking
V2BL	25	I/O	TTL, 4	V2 Horizontal/Composite Blanking
V1CLK	2	I	TTL	V1 Data Clock
V2CLK	29	I	TTL	V2 Data Clock
V1PH	158	I	TTL	V1 Phase
V2PH	24	I	TTL	V2 Phase
V1IEN*	157	O	TTL, 4	V1 Input Enable
V2IEN*	23	O	TTL, 4	V2 Input Enable
STALLRQ*	31	I	TTL	Stall Request
STALL*	30	O	TTL, 4	Stall
<b>FRAME BUFFER INTERFACE</b>				
<i>Address/Data</i>				
FBD[31:0]	99, 101:102, 105:107, 109:125, 128:136	I/O	TTL, 4	Frame Buffer Data Bus
FBA[9:0]	98:94, 92:88	O	TTL, 8	Frame Buffer Address Bus
<i>Control</i>				
RAS[1:0]*	87:86	O	TTL, 8	Row Address Strokes
CAS[1:0]*	85:84	O	TTL, 8	Column Address Strokes
WE*	83	O	TTL, 12	Write Enable
DTE*	82	O	TTL, 12	Data Transfer Enable
FRDY	81	I	TTL	FIFO Ready
SBCLK	76	O	TTL, 8	Serial Bus Clock
SOE[1:0]*	73:72	O	TTL, 8	Serial Port Output Enable
MCLK	71	I	TTL	Memory Clock
FCLK	70	O	TTL, 8	FIFO Write Clock
<b>POWER AND GROUND</b>				
VDD	18, 21, 46, 64, 75, 100, 103, 126, 140, 155	PWR	N/A	+5 VDC for Digital Logic and Interface Buffers
VSS	13, 17, 28, 47, 63, 74, 93, 104, 108, 127, 139, 154	PWR	N/A	Ground for Digital Logic and Interface Buffers

ATI019039





## 2. DETAILED SIGNAL DESCRIPTIONS

### 2.1 Host Interface — ISA

Signal	Pin	Type	Cell	Function
SAD[15:0]	48:62, 65	I/O	TTL, 4	<b>Address/Data Bus.</b> Bidirectional, multiplexed address/data bus that transfers video data and operation status and commands between the host system and the DVP.
DEN*	44	OD	TTL, 8	<b>Data Buffer Enable.</b> 0 Enables the host data bus buffer.
DDIR	43	OD	TTL, 8	<b>Data Buffer Direction.</b> Specifies the direction of data flow on SAD[15:0]. 0 The host system is reading data from SAD[15:0]; 1 The host system is writing data to SAD[15:0].
IOR*	35	I	TTL	<b>I/O Read.</b> 0 Specifies an I/O read cycle.
IOW*	36	I	TTL	<b>I/O Write.</b> 0 Specifies an I/O write cycle.
AEN	32	I	TTL	<b>Address Enable.</b> 0 I/O cycle in progress. 1 DMA cycle in progress.
NC	38	N/A	N/A	<b>No Connect.</b> (must be left floating).
NC	37	N/A	N/A	<b>No Connect.</b> (must be left floating).
IRQ	45	O	TTL, 4	<b>Interrupt Request.</b> 1 The DVP is requesting service from the host system.
CHRDY	40	OD	TTL, 24	<b>Channel Ready.</b> 0 The DVP is not ready to complete the current host access cycle. 1 The current host access cycle is complete.
IO16*	39	OD	TTL, 24	<b>16-bit I/O Cycle.</b> 0 The DVP is able to respond as a 16-bit I/O data device for both read and write cycles.
NC	41	N/A	N/A	<b>No Connect.</b> (must be left floating).
RESET	42	I	TTL	<b>Reset.</b> 1 Stops all DVP activity and resets the hardware.
NC	33	N/A	N/A	<b>No Connect.</b> (must be left floating).
AUXCS*	34		TTL	<b>Auxiliary Chip Select.</b> When programmed for aux ISA mode, primary and secondary addresses are ignored; AUXCS* and SAD[3:1] select specific registers.

ATI019040



## 2.2 Host Interface — MCA

Signal	Pin	Type	Cell	Function																																				
AD[15:0]	48:62, 65	I/O	TTL, 4	<b>Address/Data Bus.</b> Bidirectional, multiplexed address/data bus that transfers video data and operation status and commands between the host system and the DVP.																																				
DEN*	44	OD	TTL, 8	<b>Data Buffer Enable.</b> 0 Enables the host data bus buffer.																																				
DDIR	43	OD	TTL, 8	<b>Data Buffer Direction.</b> Specifies the direction of data flow on SAD[15:0]. 0 The host system is reading data from SAD[15:0]; 1 The host system is writing data to SAD[15:0].																																				
S1*	35	I	TTL	<b>Status 1.</b> Specifies current bus cycle (used with MIO* and S0*).																																				
S0*	36	I	TTL	<b>Status 0.</b> Specifies current bus cycle (used with MIO* and S1*).																																				
MIO*	32	I	TTL	<b>Memory or I/O Cycle.</b> Specifies current bus cycle current bus cycle (used with S0* and S1*): <table><tr><th>MIO*</th><th>S0*</th><th>S1*</th><th></th></tr><tr><td>0</td><td>0</td><td>0</td><td>Reserved</td></tr><tr><td>0</td><td>0</td><td>1</td><td>I/O Write</td></tr><tr><td>0</td><td>1</td><td>0</td><td>I/O Read</td></tr><tr><td>0</td><td>1</td><td>1</td><td>Inactive</td></tr><tr><td>1</td><td>0</td><td>0</td><td>Reserved</td></tr><tr><td>1</td><td>0</td><td>1</td><td>Memory Write</td></tr><tr><td>1</td><td>1</td><td>0</td><td>Memory Read</td></tr><tr><td>1</td><td>1</td><td>1</td><td>Inactive</td></tr></table>	MIO*	S0*	S1*		0	0	0	Reserved	0	0	1	I/O Write	0	1	0	I/O Read	0	1	1	Inactive	1	0	0	Reserved	1	0	1	Memory Write	1	1	0	Memory Read	1	1	1	Inactive
MIO*	S0*	S1*																																						
0	0	0	Reserved																																					
0	0	1	I/O Write																																					
0	1	0	I/O Read																																					
0	1	1	Inactive																																					
1	0	0	Reserved																																					
1	0	1	Memory Write																																					
1	1	0	Memory Read																																					
1	1	1	Inactive																																					
CDSFDBK*	38	O	TTL, 4	<b>Card Select Feedback.</b> 0 Specifies that the DVP has decoded the current address and status inputs. The DVP does not drive CDSFDBK* low during the configuration period (CDSETUP* = 0).																																				
CDSETUP*	37	I	TTL	<b>Card Setup.</b> 0 Specifies that the host system is accessing the configuration registers of the MCA adapter. To obtain adapter ID and configuration data (containing POS [Programmable Option Select] 100, 101, and 102), perform an I/O read cycle to the DVP.																																				
IRQ	45	O	TTL, 4	<b>Interrupt Request.</b> 0 The DVP is requesting service from the host system.																																				
CDCHRDY	40	OD	TTL, 24	<b>Channel Ready.</b> 1 The DVP is ready to complete the current host access cycle.																																				
CDDS16*	39	OD	TTL, 24	<b>Card Data Size.</b> 0 The DVP is able to respond as a 16-bit I/O data device for both read and write cycles.																																				



**Pixel  
Semiconductor**  
A Cirrus Logic Company

**CL-PX2070**  
**Digital Video Processor**

## 2.2 Host Interface — MCA (cont.)

Signal	Pin	Type	Cell	Function
CMD*	41	I	TTL	<b>Command.</b> 0 Valid data is on AD[15:0] (write cycle); or DVP should place valid data on AD[15:0] (read cycle).
CDRESET	42	I	TTL	<b>Reset.</b> 1 Stops all DVP activity and resets the hardware.
ADL*	33	I	TTL	<b>Address Latch.</b> 0 Demultiplexes the address from bus AD[15:0], and status from signals M/I0*, S1*, and S0*. The address and status must be valid during the low-to-high transition.
CARDEN	34	I	TTL	<b>Card Enable.</b> 1 Specifies that the data on bus AD[15:8] is valid.

## 2.3 Host Interface — Local Hardware

Signal	Pin	Type	Cell	Function
D[15:0]	48:62, 65	I/O	TTL, 4	<b>Data Bus.</b> Bidirectional data bus that transfers video data between the host system and the DVP.
RS[3:1]	34:33, 44	I	TTL	<b>Register Select.</b> Specify the register address during a host access.
NC	43	N/A	N/A	<b>No Connect.</b> (must be left floating).
IOR*	35	I	TTL	<b>I/O Read.</b> 0 Specifies an I/O read cycle.
IOW*	36	I	TTL	<b>I/O Write.</b> 0 Specifies an I/O write cycle.
CS*	32	I	TTL	<b>Chip Select.</b> 0 The host system is accessing the DVP.
NC	38	N/A	N/A	<b>No Connect.</b> (must be left floating).
NC	37	N/A	N/A	<b>No Connect.</b> (must be left floating).
IRQ	45	O	TTL, 4	<b>Interrupt Request.</b> 0 The DVP is requesting service from the host system.
CHRDY*	40	OD	TTL, 24	<b>Channel Ready.</b> 0 The DVP is ready to complete the current host access cycle.
NC	39	N/A	N/A	<b>No Connect.</b> (must be left floating).

ATI019042



### 2.3 Host Interface — Local Hardware (cont.)

Signal	Pin	Type	Cell	Function
PCLK	41	I	TTL	<b>Processor Clock.</b> Input clock that synchronizes the flow of data on bus D[15:0] during DMA data transfers.
RESET	42	I	TTL	<b>Reset.</b> 1 Stops all DVP activity and resets the hardware.

### 2.4 Graphics Overlay Interface

Signal	Pin	Type	Cell	Function
GPCLK	69	I	TTL	<b>Graphics Pixel Clock.</b> Clocks display output pixel data from the graphics controller.
GVS	67	I	TTL	<b>Graphics Vertical Sync.</b> Identifies the start of the vertical sync interval. A vertical sync pulse is generated once every field time for interlaced data, and once every frame time for non-interlaced data. Register DWU_MCR, bit GVSP specifies GVS as active high or active low.
GHS	66	I	TTL	<b>Graphics Horizontal Sync.</b> Identifies the start of the horizontal sync interval. A horizontal sync pulse is generated once for each input line. Register DWU_MCR, bit GHSP specifies GHS as active high or active low.
GBL	68	I	TTL	<b>Graphics Blanking.</b> Identifies the blanking interval. Register DWU_MCR, bit GBP specifies GBL as active high or active low.
ZC[3:0]	80:77	O	TTL, 4	<b>Zoom Control Bus</b> (used only with CL-PX2080 MediaDAC™). Specifies to the MediaDAC™ the zoom factor to be used on the current data.

ATI019043



## 2.5 Video Interface

Signal	Pin	Type	Cell	Function	
V1D[15:0]	156, 153:141, 138:137	I/O	TTL, 4	V1 (Video Port 1) Data Bus.	<i>VnD[15:0]</i> . Bidirectional data bus that transfers video data between the DVP and an external device through video port Vn.
V2D[15:0]	3:12, 14:16, 19:20, 22	I/O	TTL, 4	V2 (Video Port 2) Data Bus.	
V1VS	1	I/O	TTL, 4	V1 Vertical Sync.	<i>VnVS</i> . Identifies the start of the vertical sync interval. A vertical sync pulse is generated once every field time for interlaced data, and once every frame time for non-interlaced data. Register VIU_MCRp (bits OVSP/IVSP) specifies VnVS as active high or active low.
V2VS	27	I/O	TTL, 4	V2 Vertical Sync.	
V1HS	160	I/O	TTL, 4	V1 Horizontal Sync.	<i>VnHS</i> . Identifies the start of the horizontal sync interval; register VIU_MCRp (bits OHSP/IHSP) specifies VnHS as active high or active low.
V2HS	26	I/O	TTL, 4	V2 Horizontal Sync.	
V1BL	159	I/O	TTL, 4	V1 Horizontal/Composite Blanking.	<i>VnBL</i> . Identifies the blanking interval; register VIU_MCRp (bits OBP/IBP) specifies VnBL as active high or active low.
V2BL	25	I/O	TTL, 4	V2 Horizontal/Composite Blanking.	
V1CLK	2	I	TTL	V1 Data Clock.	<i>VnCLK</i> . Clocks bidirectional video data on bus VnD[15:0].
V2CLK	29	I	TTL	V2 Data Clock.	
V1PH	158	I	TTL, 4	V1 Phase.	<i>VnPH</i> . Controls data qualification and duplexing of video data on VnD[15:0].
V2PH	24	I	TTL	V2 Phase.	
V1IEN*	157	O	TTL, 4	V1 Input Enable.	<i>VnIEN*</i> . Specifies that the DVP is not driving bus VnD[15:0]. VnIEN* can be used as a tristate control by an external buffer connected to bus VnD[15:0].
V2IEN*	23	O	TTL, 4	V2 Input Enable.	
STALLRQ*	31	I	TTL	Stall Request. 0 Requests that the current transfer of video data on bus V2D[15:0] be suspended.	
STALL*	30	O	TTL, 4	Stall. 0 The DVP has suspended transferring data on V2D[15:0].	

ATI019044



## 2.6 Frame Buffer Interface

Signal	Pin	Type	Cell	Function
FBD[31:0]	136:128, 125:109, 107:105, 102:101, 99	I/O	TTL, 4	<b>Frame Buffer Data Bus.</b> Bidirectional data bus that transfers data between the DVP and the frame buffer.
FBA[9:0]	98:94, 92:88	O	TTL, 8	<b>Frame Buffer Address Bus.</b> Multiplexed output bus that specifies an address to the frame buffer. The row address is valid during the HIGH-to-LOW transition of signals RAS[1:0]*; the column address is valid during the high-to-low transition of CAS[1:0]*.
RAS[1:0]*	87:86	O	TTL, 8	<b>Row Address Strobes.</b> Instruct the frame buffer to latch the row address from bus FBA[9:0] during the HIGH-to-LOW transition.
CAS[1:0]*	85:84	O	TTL, 8	<b>Column Address Strobes.</b> Instruct the frame buffer to latch the column address from bus FBA[9:0] during the HIGH-to-LOW transition.
WE*	83	O	TTL, 12	<b>Write Enable.</b> Specifies a write cycle to the frame buffer.
DTE*	82	O	TTL, 12	<b>Data Transfer Enable.</b> Specifies a transfer cycle to the frame buffer (VRAMs only).
FRDY	81	I	TTL	<b>FIFO Ready.</b> (used only with CL-PX2080 MediaDAC™) Specifies that the input FIFO of the MediaDAC™ is ready to receive serial data from the frame buffer.
SBCLK	76	O	TTL, 8	<b>Serial Bus Clock.</b> Clocks serial data from the frame buffer (VRAMs only).
SOE[1:0]*	73:72	O	TTL, 8	<b>Serial Port Output Enable.</b> 0 Enable the frame-buffer serial data port output.
MCLK	71	I	TTL	<b>Memory Clock.</b> Synchronizes all frame buffer control signals.
FCLK	70	O	TTL, 8	<b>FIFO Write Clock.</b> (used only with CL-PX2080 MediaDAC™) Clocks serial data into the MediaDAC™.

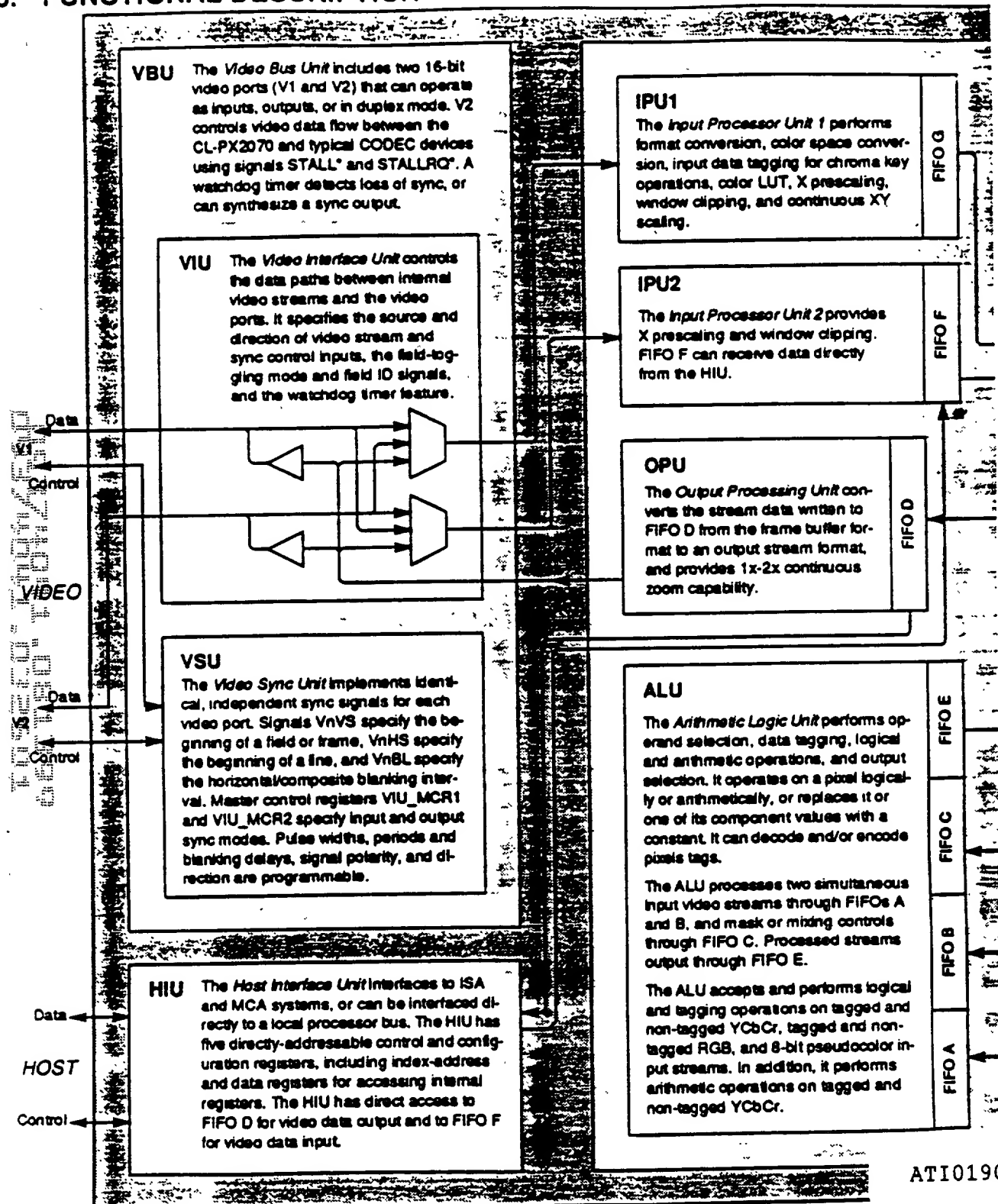
## 2.7 Power and Ground

Signal	Pin	Type	Function
VDD	18, 21, 46, 64, 75, 100, 103, 126, 140, 155	PWR	<b>+5 VDC for Digital Logic and Interface Buffers.</b> Each VDD pin must be connected directly to the VDD plane.
VSS	13, 17, 28, 47, 63, 74, 93, 104, 108, 127, 139, 154	PWR	<b>Ground for Digital Logic and Interface Buffers.</b> Each VSS pin must be connected directly to the ground plane.

ATI019045



### 3. FUNCTIONAL DESCRIPTION



ATI019046

# CL-PX2070

## Digital Video Processor



**Pixel Semiconductor**  
A Cirrus Logic Company

### VPU

The *Video Processing Unit* provides field- or frame-oriented video processing. It can simultaneously process two external, bidirectional real-time video streams and a single external, bidirectional host video or graphic data stream.

### SIU

The *Sequencer Instruction Unit* is a special-purpose microcontroller that moves pixel data between the hardware resources under the control of instruction sequences stored in the SIM.

The SIU resembles a short software loop made of conditional instructions. Each instruction causes data to move between the components, and specifies: the source of the video information, conditions for execution, destination, and the location of the next instruction. Possible sources and destinations are object buffers and FIFOs A-G.

The SIU executes SIM instructions much faster than the stream rates of typical video data. Therefore, the instructions are conditional. At any given time, the FIFO associated with the current instruction may or may not be ready to source or receive data. If it is not ready, the SIU skips the instruction and continues with the next until an instruction is found which can be executed. This instruction format also provides branching and looping capabilities.

### CL-PX2070 Digital Video Processor

The DVP combines following capabilities for multiple concurrent video data streams:

- frame buffer memory management,
- arithmetic and logical processing,
- a programmable host system bus interface,
- flexible, multistream video datapath,
- windowing control.

Possible applications of the DVP include presentation, video editing, multimedia authoring, animation, video teleconferencing, and video capture and scaling.

### OBU

The *Object Buffer Unit* allows each object buffer to be locked to either video source, or to be programmed to operate independently. Object buffers may also be placed anywhere within the linearly-addressable frame buffer. OBU registers specify the following for each object buffer: size and location, operating mode, X and Y BLT directions, FIFO association, chrominance and luminance channel masking, and output decimation.

### RFU

The *Reference Frame Unit* provides simultaneous access to eight object buffers and four display windows

### MMU

The *Memory Management Unit* provides DRAM/VRAM support and translates the parameters from the rest of the system into physical memory address using register MMU\_MCR. Frame buffer size can be up to 8 megabytes.

### DWU

The *Display Window Unit* allows each display window to be any size or location. When used with the CL-PX2080, display windows can overlap.

**GRAPHICS OVERLAY**

**FRAME BUFFER**

Data

Control

Control

ATI019047





## 4. DETAILED REGISTER DESCRIPTIONS

This section lists and defines the CL-PX2070 DVP registers.

**NOTE:** In order to maintain compatibility with future Pixel Semiconductor products, all reserved registers bits must be written as '0'. Data values in reserved register locations are not guaranteed on readback.

Register names containing lower-case variables represent groups of registers with similar functions. Refer to the *Conventions* table on page 8 for a list of DVP register variables.

### 4.1 HIU: Host Interface Unit

**Table 4-1. HIU Register Address Map**

Register	Pri. Map	Sec. Map	Definition	Used by Registers		Ref. Section
HIU_0	27C0	0290	Register VO Address 0	HIU_CSU	Configuration Setup	4.1.1, p. 23
				HIU_DBG	Debug Control	4.1.2, p. 24
				HIU_DRD	Debug Read	4.1.3, p. 24
HIU_1	27C2	0292	Register VO Address 1	HIU_OCS	Operation Control/Status	4.1.5, p. 26
				HIU_IRQ	Interrupt Request	4.1.4, p. 25
HIU_2	27C4	0294	Register VO Address 2	HIU_RIN	Register Index	4.1.6, p. 27
HIU_3	27C6	0296	Register VO Address 3	HIU_RDT	Register Data Port	4.1.7, p. 28
HIU_4	27C8	0298	Register VO Address 4	HIU_MDT	Memory Data Port	4.1.8, p. 28

**Table 4-2. HIU Registers Accessed by the Register Data Port**

Register	Index	Definition	Ref. Section
HIU_ISU	0001	Interrupt Setup	4.1.9, p. 29



#### 4.1.1 HIU\_CSU: Configuration Setup

I/O Address    27C0 (Primary Map)  
                  0290 (Secondary Map)

HIU\_CSU is a read-only register that stores hardware configuration data for the DVP. An external configuration register must provide configuration data to bits 5:0 during the reset interval. HIU\_CSU is shadowed by registers HIU\_DBG and HIU\_DRD.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				VER			RSVD			HSB		RSVD	FBT	PAS	

Bit #	Access	Reset	Description
15:12	R	0000	RSVD    Reserved (read as '0').
11:8	R	0000	VER     DVP Device Version 0000    CL-PX2070 revision AB 0001    CL-PX2070, revision AC
7:6	R	00	RSVD    Reserved (read as '00')
5:3	R	111	HSB     Host System Bus. Specifies the type of host system connected to the DVP. 000     ISA bus 001     MCA bus 010     Reserved 011     Local hardware interface 100     Aux ISA 101     Aux MCA 111     Local hardware interface XXX     All other configurations reserved
2	R	1	RSVD    Reserved (read as 1)
1	R	1	FBT     Frame Buffer Jumper State. (Used only for software configuration. Does not affect internal DVP operation.) 0        DRAM 1        VRAM
0	R	0	PAS     Port Address Select. Specifies the I/O address map that the host system should use when accessing the DVP. 0        Primary port map 1        Secondary port map

ATI019049



#### 4.1.2 HIU\_DBG: Debug Control

I/O Address 27C0 (Primary Map)

0290 (Secondary Map)

HIU\_DBG is a write-only register that controls the diagnostic mode of the DVP. Register HIU\_OCS, field MDE enables access to this register when set to '1.' HIU\_DBG is shadowed by register HIU\_DRD.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						DRE	RSVD								

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:10	W	0h	RSVD Reserved (read as '0').
-------	---	----	------------------------------

9	W	0	DRE Debug Read Enable. Enables access to shadow register HIU_DRD. 0 Disable debug read 1 Enable debug read
---	---	---	--

8:0	W	00h	RSVD Reserved (read as '0').
-----	---	-----	------------------------------

#### 4.1.3 HIU\_DRD: Debug Read

I/O Address 27C0 (Primary Map)

0290 (Secondary Map)

See also: HIU\_DBG: Debug Control, p. 24

SIU\_MCR: SIU Master Control, p. 58

HIU\_OCS: Operation Control/Status, p. 26

SIUs\_SIM: Sequencer Instruction Memory, p. 61

HIU\_DRD is a read-only register that provides diagnostic information, including the global Error Detection Trap, the current object buffer counters, and the SIU current index. HIU\_DRD is a shadow register to HIU\_CSU. Read access to this register is enabled when HIU\_OCS, field MDE and HIU\_DBG, field DRE are set to '1.'

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDT		XC				YC				SIMIN					

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15	R	0	EDT Error Detection Trap. This field is the logical OR of all FIFO overflow and underflow flags, and the watchdog timeout. 0 No error 1 Error detected
----	---	---	--

14:10	R	0h	XC X Counter. Upper 5 bits of X Counter (Single-Step Mode). (0-1Fh)
-------	---	----	---

9:5	R	0h	YC Y Counter. Upper 5 bits of Y Counter (Single-Step Mode). (0-1Fh)
-----	---	----	---

4:0	R	0h	SIMIN Sequence Instruction Memory Current Index (0-1Fh)
-----	---	----	---

ATI019050



#### 4.1.4 HIU\_IRQ: Interrupt Request

I/O Address    27C2 (Primary Map)  
                  0292 (Secondary Map)

See also:        HIU\_OCS: Operation Control/Status, p. 26  
                  HIU\_ISU: Interrupt Setup, p. 29

HIU\_IRQ is a read-only register that accesses all interrupt requests generated by the IPU1, IPU2, OBU, the watchdog timer, and the FIFO overflow and underflow flags. An interrupt service routine typically uses HIU\_IRQ to determine the interrupt request source(s). HIU\_IRQ shadows register HIU\_OCS. HIU\_OCS, field SRC must be set to '1' to enable this register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										OBT	IP2C	IP1C	FUN	FOV	WDT

Bit #	Access	Reset	Description
15:6	R	0h	RSVD    Reserved (read as '0').
5	R	0	OBT    Object Buffer Termination (auto reset on read). 0       No interrupt request 1       Specifies that an object buffer termination condition occurred in the OBU.
4	R	0	IP2C    IPU2 Counter (auto reset on read). 0       No interrupt request 1       Specifies that a line, field, or vertical sync pulse interrupt request occurred in the IPU2.
3	R	0	IP1C    IPU1 Counter (auto reset on read). 0       No interrupt request 1       Specifies that a line, field, or vertical sync pulse interrupt request occurred in the IPU1.
2	R	0	FUN    FIFO Underflow (auto reset on read). 0       No interrupt request 1       Specifies that an underflow condition occurred in a FIFO. (See SIU_FOU: FIFO Overflow/Underflow, p. 60.)
1	R	0	FOV    FIFO Overflow (auto reset on read). 0       No interrupt request 1       Specifies that an overflow condition occurred in a FIFO. (See SIU_FOU: FIFO Overflow/Underflow, p. 60.)
0	R	0	WDT    Watchdog Timer to generate signal IRQ (auto reset on read). 0       No interrupt request 1       Specifies that a timeout condition occurred in VIU_WDT.

AT1019051



#### 4.1.5 HIU\_OCS: Operation Control/Status

I/O Address    27C2h (Primary Map)  
                  0292 (Secondary Map)

Register HIU\_OCS controls the operating mode of the DVP and provides status indicators. HIU\_OCS is shadowed during read cycles by register HIU\_IRQ.

**NOTE:** Modifications to registers designated as posted do not affect the operation of the DVP until a post command is issued either manually using bit PMC, or automatically by the SIU. Automatic posting typically occurs between field or frame times.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	FDNE	FFNF	RSVD	SRC	MDE	DPC	MPC	PMC	RSVD	SR					IEM

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15	R	0	RSVD	Reserved (read as '0').
14	R	0	FDNE	FIFO D Nearly Empty. 1      FIFO D is within 16 pixels of being empty
13	R	0	FFNF	FIFO F Nearly Full. 1      FIFO F is within 16 pixels of being full
12	R	0	RSVD	Reserved (read as '0').
11	R/W	0	SRC	Status Read Select. Specifies register to access during a read cycle. 0      Read status from register HIU_OCS 1      Read status from shadow register HIU_IRQ
10	R/W	0	MDE	Master Debug Enable. 0      Disable debug support registers HIU_DBG and HIU_DRD 1      Enable access to registers HIU_DBG and HIU_DRD
9	R/W	0	DPC	Display Window Posting Operation Control (auto reset). Enables the register posting mode of the DWU. 0      Disable posting 1      Enable posting (auto reset on post)
8	R/W	0	MPC	Master Posting Control (auto reset). Enables all DVP register posting logic. 0      Disable posting 1      Enable posting (auto reset on post)
7	R/W	0	PMC	Posting Mode Control. 0      Specifies normal register posting operation (waits for vertical sync) 1      Forces immediate post all registers (DPC, MPC must = '1')
6:5	R/W	0	RSVD	Reserved (read as '0').

ATI019052



Bit #	Access	Reset	Description
4	R/W	0	SR Soft Reset. Causes a soft reset to be performed on all internal units. All registers are reset to 0, all FIFOs are cleared, and all counters are set to 0. Output signals are not placed in three-state. 0 No reset performed 1 Perform soft reset
3:0	R/W	0000	IEM Interrupt Enable Mask. Enables interrupt requests. When more than one interrupt source is enabled, the requests are ORed — any source can assert signal IRQ. See Section 4.1.9 on page 29 for additional information on the interrupt system. 0001 Enable counter to generate signal IRQ 0010 Enable watchdog to generate signal IRQ 0100 Enable object buffer termination to generate signal IRQ 1000 Enable FIFO overflow/underflow to generate signal IRQ

#### 4.1.6 HIU\_RIN: Register Index

I/O Address 27C4 (Primary Map)  
0294 (Secondary Map)

Register HIU\_RIN specifies the index value of the next register to be accessed. An optional control (bit AIC) automatically increments the index address on consecutive read or write cycles.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AIC	RIN														

Bit #	Access	Reset	Description
15	R/W	0h	AIC Automatic Increment Control (index address). 0 Disable 1 Enable
14:0	R/W	0h	RIN Register Index. (0-7FFFh)

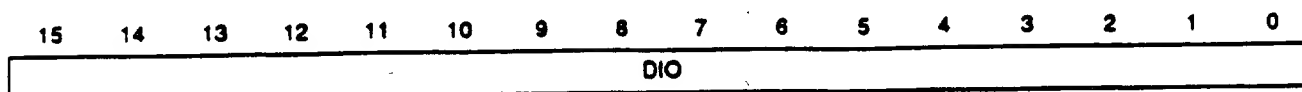
ATI019053



#### 4.1.7 HIU\_RDT: Register Data Port

I/O Address 27C6 (Primary Map)  
0296 (Secondary Map)

HIU\_RDT is the register data port. Registers are index-mapped to HIU\_RDT by HIU\_RIN.



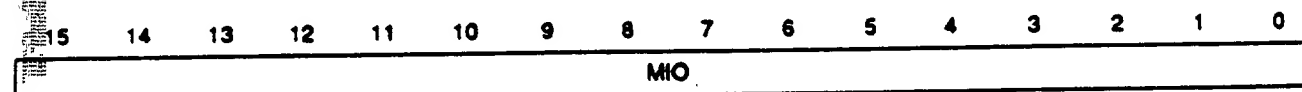
Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:0	R/W	0h	DIO Register Data I/O
------	-----	----	-----------------------

#### 4.1.8 HIU\_MDT: Memory Data Port

I/O Address 27C8 (Primary Map)  
0298 (Secondary Map)

I/O port HIU\_MDT accesses the frame buffer. To maintain data integrity when reading or writing to this port, first check the status of the appropriate FIFO.



Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:0	R/W	0h	MIO Memory Data I/O
------	-----	----	---------------------

ATI019054



#### 4.1.9 HIU\_ISU: Interrupt Setup

I/O Address HIU\_RDT  
Index 0001

Register HIU\_ISU specifies the interrupt modes for the IPU1, IPU2, and the OBU. Any interrupt requests generated in the IPU1, IPU2, and OBU must also be enabled through register HIU\_OCS, field IEM.

IPU interrupts are combined with an AND function. If more than one interrupt source is enabled within an IPnS field, all sources must be active before an interrupt request is posted.

The interrupt sources in the OBIS field use an OR function. If more than one interrupt source is selected, any one active source can trigger an interrupt.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		IP2S			IP1S			OBIS							

Bit #	Access	Reset	Description
15:14	R/W	00	RSVD Reserved (read as '0').
13:11	R/W	000	IP2S IPU2 Interrupt Select. Specifies the IPU2 line count, field count, and input vertical sync pulse combination required to generate an interrupt request. 001 Interrupt on line count 010 Interrupt on field count 100 Interrupt on vertical sync
10:8	R/W	000	IP1S IPU1 Interrupt Select. Specifies the IPU1 line count, field count, and input vertical sync pulse combination required to generate an interrupt request. 001 Interrupt on line count 010 Interrupt on field count 100 Interrupt on vertical sync
7:0	R/W	0h	OBIS Object Buffer Termination Interrupt Request. Specifies the OBU object buffer termination conditions combination required to generate interrupt request signal IRQ. 01h Object buffer 0 termination 02h Object buffer 1 termination 04h Object buffer 2 termination 08h Object buffer 3 termination 10h Object buffer 4 termination 20h Object buffer 5 termination 40h Object buffer 6 termination 80h Object buffer 7 termination

AT1019055





## 4.2 VBU: Video Bus Unit

Register	Index	Definition	Posted?	Ref. Section
<b>VIU: Video Interface Unit</b>				4.2.1, p. 30
VIU_MCR1	1000	VIU Master Control V1	—	4.2.1.1, p. 30
VIU_MCR2	1001	VIU Master Control V2	—	4.2.1.1, p. 30
VIU_DPC1	1002	Datapath Control, Field 1	POSTED	4.2.1.2, p. 32
VIU_DPC2	1003	Datapath Control, Field 2	POSTED	4.2.1.2, p. 32
VIU_WDT	1004	Watchdog Timer	POSTED	4.2.1.3, p. 33
VIU_TEST	1006	Test Register	—	4.2.1.4, p. 34

## **VSU: Video Sync Unit** 4.2.2, p. 35

VSU_HSW	1100	Horizontal Sync Width	POSTED	4.2.2.1, p. 35
VSU_HAD	1101	Horizontal Active Delay	POSTED	4.2.2.2, p. 36
VSU_HAP	1102	Horizontal Active Pixels	POSTED	4.2.2.3, p. 36
VSU_HP	1103	Horizontal Period	POSTED	4.2.2.4, p. 36
VSU_VSW	1104	Vertical Sync Width	POSTED	4.2.2.5, p. 37
VSU_VAD	1105	Vertical Active Delay	POSTED	4.2.2.6, p. 37
VSU_VAP	1106	Vertical Active Pixels	POSTED	4.2.2.7, p. 38
VSU_VP	1107	Vertical Period	POSTED	4.2.2.8, p. 38

### 4.2.1 VIU: Video Interface Unit

#### 4.2.1.1 VIU\_MCRp: VIU Master Control

I/O Address HIU\_RDT  
Index 1000 (VIU\_MCR1: VIU Master Control V1) 1001 (VIU\_MCR2: VIU Master Control V2)

Registers VIU\_MCR1 and VIU\_MCR2 specify the functional and I/O characteristics of Video Port Interfaces 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STM	OPF	OSS	OVSP	OHSP	OBP	OBT	IFP	ISS	IVSP	IHSP	IBP	IBT	IOM		

ATI019056

**CL-PX2070**  
*Digital Video Processor*

[illegible]



#### 4.2.1.2 VIU\_DPCf: Datapath Control

POSTED

I/O Address HIU\_RDT

Index 1002 (VIU\_DPC1: Datapath Control, Field 1)

1003 (VIU\_DPC2: Datapath Control, Field 2)

Registers VIU\_DPC1 and VIU\_DPC2 specify the flow of stream data and the source of control sync references for the IPU1, the IPU2, and the OPU for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD				VSUDC			IPU1DC			IPU2DC			ODC		

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:12	R/W	0000	RSVD Reserved (read as '0').
-------	-----	------	------------------------------

11:9	R/W	000	VSUDC VSU Datapath Control
		000	V1 sources clock
		001	V1 sources clock, V1PH qualified
		010	V2 sources clock
		011	V2 sources clock, V2PH qualified
		100	MCLK+3 (sequencer clock) timebase
		101	MCLK+6 timebase
		XXX	All other configurations reserved

8:6	R/W	000	IPU1DC IPU1 Datapath Control. Specifies the source of control sync references and input stream data for the IPU1.
		000	V1 sources sync and data
		001	V1 sources sync and data, V1PH qualified
		010	V2 sources sync and data
		011	V2 sources sync and data, V2PH qualified
		100	OPU sources data, MCLK+3 HS timebase, VSU sources sync
		101	OPU sources data, MCLK+6 HS timebase, VSU sources sync
		XXX	All other configurations reserved

5:3	R/W	000	IPU2DC IPU2 Datapath Control. Specifies the source of control sync references and input stream data for the IPU2.
		000	V1 sources sync and data
		001	V1 sources sync and data, V1PH qualified
		010	V2 sources sync and data
		011	V2 sources sync and data, V2PH qualified
		100	OPU sources data, MCLK+3 HS timebase, VSU sources sync
		101	OPU sources data, MCLK+6 HS timebase, VSU sources sync
		110	HIU sources data directly to FIFO F, no sync controls
		111	Reserved

ATI019058



Bit #	Access	Reset	Description
2.0	R/W	000	ODC OPU Datapath Control. Specifies the source of control sync references and the destination of output stream data from the OPU.
		000	V1 sources sync
		001	V1 sources sync, V1PH qualified
		010	V2 sources sync
		011	V2 sources sync, V2PH qualified
		100	VSU sources sync, MCLK+3 timebase
		101	VSU sources sync, MCLK+6 timebase
		110	HIU receives data directly from FIFO D, no sync controls
		111	Reserved

#### 4.2.1.3 VIU\_WDT: Watchdog Timer

POSTED

I/O Address HIU\_RDT

Index 1004

Register VIU\_WDT controls watchdog timer operation, and specifies the field toggle mode of the SIU.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	MMS	MFTS			WTE	TMOUT									

Bit #	Access	Reset	Description
15	R/W	0	RSVD Reserved (read as '0').
14	R/W	0	MMS Manual Mode Start. Writing 0, then 1 while MFTS is programmed to 6h initiates a field toggle in manual mode.
13:11	R/W	000	MFTS Master Field Toggle Select. Specifies the field toggle mode for the SIU. The field toggles on the leading edge of vertical sync.
		000	Field timing from V1VS input
		001	Field timing from V1VS output
		010	Field timing from V2VS input
		011	Field timing from V2VS output
		100	Field timing from watchdog timer
		101	Field timing from VSU vertical sync
		110	Field timing from manual mode start
		111	Reserved
10	R/W	0	WTE Watchdog Timer Enable.
		0	Disable watchdog timer
		1	Enable watchdog timer
9:0	R/W	0h	TMOUT Timeout. Specifies the watchdog timer interval. The timebase interval is MCLK prescaled by a factor of 49,152 ( $3 \cdot 2^{14}$ ). (0-3FFh)

ATI019059



#### 4.2.1.4 VIU\_TEST: Test Register

I/O Address HIU\_RDT  
Index 1006

VIU\_TEST is a read-only test register for diagnostic use and software debugging. It allows user to monitor conditions between IPU1, IPU2, OPU, and VIU.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MF	MFID	RSVD			OBIN	OVS	OHS	OBL	OFID	I2VS	I2BL	I2FID	I1VS	I1BL	I1FID

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15	R	0	MF	Master Field. Specifies which SIU loop is being executed. 0 SIU_MCR, field SI1 1 SIU_MCR, field SI2
14	R	0	MFID	Master Field ID. Monitors the state of the VIU_WDT Master Field Toggle Select condition. Inverted from selected source field ID. For example, if V1 and IPU1 are selected, this bit is inverted from I1FID. See Section 4.2.1.3 on page 33.
13:11	R	000	RSVD	Reserved.
10	R	0	OBIN	Blank in from OPU based on the clipping values programmed into registers OPU_XBI1, OPU_XEI1, OPU_YBI1, and OPU_YEI1.
9	R	0	OVS	OPU Vertical Sync.
8	R	0	OHS	OPU Horizontal Sync.
7	R	0	OBL	OPU Blank.
6	R	0	OFID	OPU Field ID. Value depends on OPU field polarity (specified by OPU_MCRf, bit FPS).
5	R	0	I2VS	IPU2 Vertical Sync.
4	R	0	I2BL	IPU2 Blank.
3	R	0	I2FID	IPU2 Field ID. Value depends on IPU2 field polarity (specified by IPU2_MCRf, bit FPS).
2	R	0	I1VS	IPU1 Vertical Sync.
1	R	0	I1BL	IPU1 Blank.
0	R	0	I1FID	IPU1 Field ID. Value depends on IPU1 field polarity (specified by IPU1_MCRf, bit FPS).

ATI019060



#### 4.2.2 VSU: Video Sync Unit

The following sections describe the VSU registers, shown in Figure 4-1 and Figure 4-2.

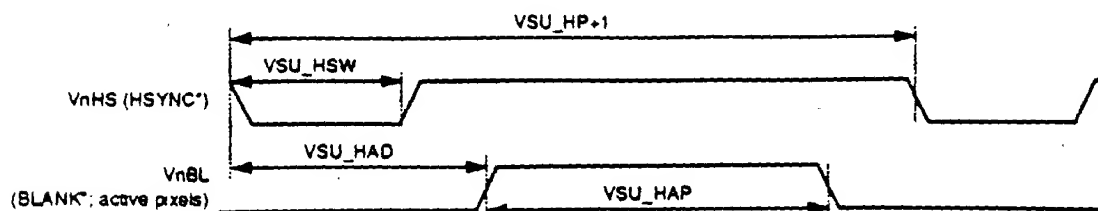


Figure 4-1. VSU Horizontal Sync Timing

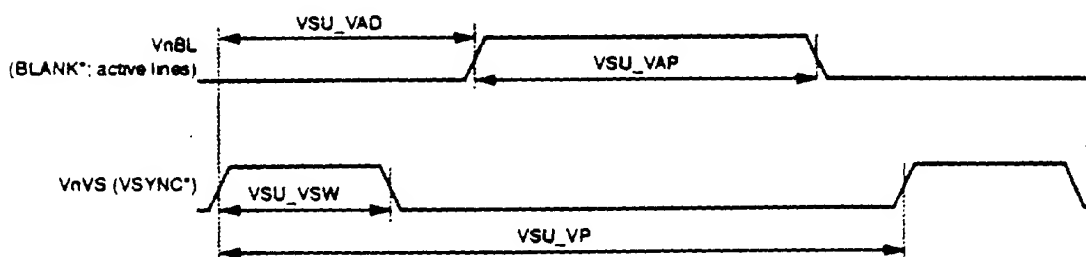


Figure 4-2. VSU Vertical Sync Timing

##### 4.2.2.1 VSU\_HSW: Horizontal Sync Width

POSTED

I/O Address HIU\_RDT

Index 1100

Register VSU\_HSW specifies the width of the horizontal sync pulse generated by the internal sync generator. The timebase is specified by registers VIU\_DPCf, bits IPU1DC and IPU2DC.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD									HSW						

Bit #	Access	Reset	Description
15:7	R/W	0h	RSVD Reserved (read as '0').
6:0	R/W	0h	HSW Horizontal Sync Width. (0-7Fh) (20h - 7Fh in loopback mode)

ATI012061



#### 4.2.2.2 VSU\_HAD: Horizontal Active Delay

POSTED

I/O Address HIU\_RDT

Index 1101

Register VSU\_HAD specifies the delay from the start of the horizontal sync pulse generated by the internal sync generator to the beginning of the horizontal active interval. The timebase is specified by VIU\_DPCf, bits IPU1DC and IPU2DC. VSU\_HAD must equal VSU\_HSW+3 when OPU\_MCRf, bit LSM = 1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						HAD									

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:10	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

9:0	R/W	0h	HAD Horizontal Active Delay. (0-3FFh)
-----	-----	----	---------------------------------------

#### 4.2.2.3 VSU\_HAP: Horizontal Active Pixels

POSTED

I/O Address HIU\_RDT

Index 1102

Register VSU\_HAP specifies the width of the horizontal active interval generated by the internal sync generator. The timebase is input memory clock signal MCLK prescaled by a factor of 3 or 6, as specified by VIU\_DPCf, bits IPU1DC and IPU2DC.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						HAP									

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:11	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

10:0	R/W	0h	HAP Horizontal Active Pixels (0-3FFh)
------	-----	----	---------------------------------------

#### 4.2.2.4 VSU\_HP: Horizontal Period

POSTED

I/O Address HIU\_RDT

Index 1103

Register VSU\_HP specifies the width of the horizontal sync period generated by the internal sync generator. The timebase is input memory clock signal MCLK prescaled by a factor of 3 or 6, as specified by VIU\_DPCf, bits IPU1DC and IPU2DC.

ATI019062

**CL-PX2070**  
Digital Video Processor



**Pixel Semiconductor**  
A Cirrus Logic Company

**NOTE:** The number entered in HP must be one less than the desired interval.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										HP					

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:10	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

9:0	R/W	0h	HP Desired Horizontal Period = (0-3FFh) - 1
-----	-----	----	---

#### 4.2.2.5 VSU\_VSW: Vertical Sync Width

POSTED

I/O Address HIU\_RDT

Index 1104

Register VSU\_VSW specifies the width of the vertical sync pulse generated by the internal sync generator. The timebase is the horizontal sync interval specified by register VSU\_HP.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										VSW					

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:7	R/W	0h	RSVD Reserved (read as '0').
------	-----	----	------------------------------

6:0	R/W	0h	VSW Vertical Sync Width (0-7Fh)
-----	-----	----	---------------------------------

#### 4.2.2.6 VSU\_VAD: Vertical Active Delay

POSTED

I/O Address HIU\_RDT

Index 1105

Register VSU\_VAD specifies the delay from the start of the vertical sync pulse generated by the internal sync generator to the beginning of the vertical active interval. The timebase is the horizontal sync interval specified by register VSU\_HP.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD										VAD					

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:10	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

9:0	R/W	0h	VAD Vertical Active Delay. (0-3FFh)
-----	-----	----	-------------------------------------

AT1019063





#### 4.2.2.7 VSU\_VAP: Vertical Active Pixels

POSTED

I/O Address HIU\_RDT

Index 1106

Register VSU\_VAP specifies the width of the vertical active interval generated by the internal sync generator. The timebase is the horizontal sync interval specified by register VSU\_HP.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						VAP									

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:11	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

10:0	R/W	0h	VAP Vertical Active Pixels. (0-7FFh)
------	-----	----	--------------------------------------

#### 4.2.2.8 VSU\_VP: Vertical Period

POSTED

I/O Address HIU\_RDT

Index 1107

Register VSU\_VP specifies the width of the vertical sync period generated by the internal sync generator. The timebase is the horizontal sync interval specified by register VSU\_HP. This register also provides the enable and single sweep controls for the internal sync generator.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SGE	SSE	VFL	RSVD	VP											

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15	R/W	0	SGE Sync Generator Enable. (Enabled when SSE = 1.) 1 Another single sweep occurs (SGE resets to '0' at the end of the sweep)
----	-----	---	---

14	R/W	0	SSE Single Sweep Enable. Enables single sweep mode. 0 SGE ignored 1 SGE enabled
----	-----	---	---

13	R/W	0	VFL Video Field Lock. 0 No field lock 1 Field-locks (synchronizes) VSU to the incoming field of the video source selected as the master in register VIU_WDT, bit MFTS; allows an internal process that may run much faster to remain in sync with an incoming stream.
----	-----	---	---

12	R/W	0000	RSVD Reserved (read as '0').
----	-----	------	------------------------------

9:0	R/W	0h	VP Vertical Active Count. (0-7FFh)
-----	-----	----	------------------------------------

ATI019064



#### 4.3 VPU: Video Processor Unit

Name	Index	Definition	Posted?	Ref. Section
VPU Global Control				4.3.1, p. 44
VPU_MCR	2000	VPU Master Control	POSTED	4.3.1.1, p. 44
IPU1: Input Processor Unit 1				4.3.2, p. 45
IPU1_PIX	2100	Pixel Count	—	4.3.2.1, p. 45
IPU1_LIC	2101	Line Count	—	4.3.2.2, p. 45
IPU1_FLC	2102	Field Count	—	4.3.2.3, p. 46
IPU1_LIR	2103	Line Count Interrupt Request	—	4.3.2.4, p. 46
IPU1_FIR	2104	Field Count Interrupt Request	—	4.3.2.5, p. 46
IPU1_LRB	2200	LUT RAM Base Address	—	4.3.2.6, p. 47
IPU1_LRD	2201	LUT RAM Data	—	4.3.2.7, p. 47
IPU1_MCR1	3000	IPU1 Master Control, Field 1	POSTED	4.3.2.8, p. 48
IPU1_XBF1	3001	X Begin Fraction, Field 1	POSTED	4.3.2.9, p. 49
IPU1_XBI1	3002	X Begin Integer, Field 1	POSTED	4.3.2.9, p. 49
IPU1_XEI1	3003	X End Integer, Field 1	POSTED	4.3.2.10, p. 50
IPU1_XSF1	3004	X Shrink Fraction, Field 1	POSTED	4.3.2.11, p. 50
IPU1_XSI1	3005	X Shrink Integer, Field 1	POSTED	4.3.2.11, p. 50
IPU1_YBF1	3006	Y Begin Fraction, Field 1	POSTED	4.3.2.12, p. 51
IPU1_YBI1	3007	Y Begin Integer, Field 1	POSTED	4.3.2.12, p. 51
IPU1_YEI1	3008	Y End Integer, Field 1	POSTED	4.3.2.13, p. 51
IPU1_YSF1	3009	Y Shrink Fraction, Field 1	POSTED	4.3.2.14, p. 52
IPU1_YSI1	300a	Y Shrink Integer, Field 1	POSTED	4.3.2.14, p. 52
IPU1_KFC1	300b	Key Function Code, Field 1	POSTED	4.3.2.15, p. 52
IPU1_MMY1	300c	Chroma Key Y/R Max/Min, Field 1	POSTED	4.3.2.16, p. 53
IPU1_MMU1	300d	Chroma Key U/G Max/Min, Field 1	POSTED	4.3.2.16, p. 53
IPU1_MMV1	300e	Chroma Key V/B Max/Min, Field 1	POSTED	4.3.2.16, p. 53
IPU1_MCR2	3100	IPU1 Master Control, Field 2	POSTED	4.3.2.8, p. 48
IPU1_XBF2	3101	X Begin Fraction, Field 2	POSTED	4.3.2.9, p. 49
IPU1_XBI2	3102	X Begin Integer, Field 2	POSTED	4.3.2.9, p. 49



### 4.3 VPU: Video Processor Unit (cont.)

Name	Index	Definition	Posted?	Ref. Section
IPU1_XEI2	3103	X End Integer, Field 2	POSTED	4.3.2.10, p. 50
IPU1_XSF2	3104	X Shrink Fraction, Field 2	POSTED	4.3.2.11, p. 50
IPU1_XSI2	3105	X Shrink Integer, Field 2	POSTED	4.3.2.11, p. 50
IPU1_YBF2	3106	Y Begin Fraction, Field 2	POSTED	4.3.2.12, p. 51
IPU1_YBI2	3107	Y Begin Integer, Field 2	POSTED	4.3.2.12, p. 51
IPU1_YEI2	3108	Y End Integer, Field 2	POSTED	4.3.2.13, p. 51
IPU1_YSF2	3109	Y Shrink Fraction, Field 2	POSTED	4.3.2.14, p. 52
IPU1_YSI2	310a	Y Shrink Integer, Field 2	POSTED	4.3.2.14, p. 52
IPU1_KFC2	310b	Key Function Code, Field 2	POSTED	4.3.2.15, p. 52
IPU1_MMY2	310c	Chroma Key Y/R Max/Min, Field 2	POSTED	4.3.2.16, p. 53
IPU1_MMU2	310d	Chroma Key U/G Max/Min, Field 2	POSTED	4.3.2.16, p. 53
IPU1_MMV2	310e	Chroma Key V/B Max/Min, Field 2	POSTED	4.3.2.16, p. 53
<b>IPU2: Input Processor Unit 2</b>				4.3.3, p. 54
IPU2_PIX	2300	Pixel Count	—	4.3.3.1, p. 54
IPU2_LIC	2301	Line Count	—	4.3.3.2, p. 54
IPU2_FLC	2302	Field Count	—	4.3.3.3, p. 54
IPU2_LIR	2303	Line Count Interrupt Request	—	4.3.3.4, p. 55
IPU2_FIR	2304	Field Count Interrupt Request	—	4.3.3.5, p. 55
IPU2_MCR1	3200	IPU2 Master Control, Field 1	POSTED	4.3.3.6, p. 56
IPU2_XBI1	3202	X Begin Integer, Field 1	POSTED	4.3.3.7, p. 56
IPU2_XEI1	3203	X End Integer, Field 1	POSTED	4.3.3.8, p. 57
IPU2_YBI1	3207	Y Begin Integer, Field 1	POSTED	4.3.3.9, p. 57
IPU2_YEI1	3208	Y End Integer, Field 1	POSTED	4.3.3.10, p. 58
IPU2_MCR2	3300	IPU2 Master Control, Field 2	POSTED	4.3.3.6, p. 56
IPU2_XBI2	3302	X Begin Integer, Field 2	POSTED	4.3.3.7, p. 56
IPU2_XEI2	3303	X End Integer, Field 2	POSTED	4.3.3.8, p. 57
IPU2_YBI2	3307	Y Begin Integer, Field 2	POSTED	4.3.3.9, p. 57
IPU2_YEI2	3308	Y End Integer, Field 2	POSTED	4.3.3.10, p. 58



#### 4.3 VPU: Video Processor Unit (cont.)

Name	Index	Definition	Posted?	Ref. Section
SIU: Sequencer Instruction Unit			—	4.3.4, p. 58
SIU_MCR	2800	SIU Master Control	—	4.3.4.1, p. 58
SIU_FCS	2801	FIFO Control/Status	—	4.3.4.2, p. 59
SIU_FOU	2802	FIFO Overflow/Underflow	—	4.3.4.3, p. 60
SIU_FAR	4001	FIFO Auto Reset	—	4.3.4.5, p. 62
SIU0_SIM	2e00	Sequencer Instruction Memory 0	—	4.3.4.4, p. 61
SIU1_SIM	2e01	Sequencer Instruction Memory 1	—	4.3.4.4, p. 61
SIU2_SIM	2e02	Sequencer Instruction Memory 2	—	4.3.4.4, p. 61
SIU3_SIM	2e03	Sequencer Instruction Memory 3	—	4.3.4.4, p. 61
SIU4_SIM	2e04	Sequencer Instruction Memory 4	—	4.3.4.4, p. 61
SIU5_SIM	2e05	Sequencer Instruction Memory 5	—	4.3.4.4, p. 61
SIU6_SIM	2e06	Sequencer Instruction Memory 6	—	4.3.4.4, p. 61
SIU7_SIM	2e07	Sequencer Instruction Memory 7	—	4.3.4.4, p. 61
SIU8_SIM	2e08	Sequencer Instruction Memory 8	—	4.3.4.4, p. 61
SIU9_SIM	2e09	Sequencer Instruction Memory 9	—	4.3.4.4, p. 61
SIU10_SIM	2e0a	Sequencer Instruction Memory 10	—	4.3.4.4, p. 61
SIU11_SIM	2e0b	Sequencer Instruction Memory 11	—	4.3.4.4, p. 61
SIU12_SIM	2e0c	Sequencer Instruction Memory 12	—	4.3.4.4, p. 61
SIU13_SIM	2e0d	Sequencer Instruction Memory 13	—	4.3.4.4, p. 61
SIU14_SIM	2e0e	Sequencer Instruction Memory 14	—	4.3.4.4, p. 61
SIU15_SIM	2e0f	Sequencer Instruction Memory 15	—	4.3.4.4, p. 61
SIU16_SIM	2e10	Sequencer Instruction Memory 16	—	4.3.4.4, p. 61
SIU17_SIM	2e11	Sequencer Instruction Memory 17	—	4.3.4.4, p. 61
SIU18_SIM	2e12	Sequencer Instruction Memory 18	—	4.3.4.4, p. 61
SIU19_SIM	2e13	Sequencer Instruction Memory 19	—	4.3.4.4, p. 61
SIU20_SIM	2e14	Sequencer Instruction Memory 20	—	4.3.4.4, p. 61
SIU21_SIM	2e15	Sequencer Instruction Memory 21	—	4.3.4.4, p. 61
SIU22_SIM	2e16	Sequencer Instruction Memory 22	—	4.3.4.4, p. 61
SIU23_SIM	2e17	Sequencer Instruction Memory 23	—	4.3.4.4, p. 61



### 4.3 VPU: Video Processor Unit (cont.)

Name	Index	Definition	Posted?	Ref. Section
SIU24_SIM	2e18	Sequencer Instruction Memory 24	—	4.3.4.4, p. 61
SIU25_SIM	2e19	Sequencer Instruction Memory 25	—	4.3.4.4, p. 61
SIU26_SIM	2e1a	Sequencer Instruction Memory 26	—	4.3.4.4, p. 61
SIU27_SIM	2e1b	Sequencer Instruction Memory 27	—	4.3.4.4, p. 61
SIU28_SIM	2e1c	Sequencer Instruction Memory 28	—	4.3.4.4, p. 61
SIU29_SIM	2e1d	Sequencer Instruction Memory 29	—	4.3.4.4, p. 61
SIU30_SIM	2e1e	Sequencer Instruction Memory 30	—	4.3.4.4, p. 61
SIU31_SIM	2e1f	Sequencer Instruction Memory 31	—	4.3.4.4, p. 61
<b>ALU: Arithmetic and Logic Unit</b>				4.3.4.5, p. 62
ALU_MCR1	2900	ALU Master Control, Field 1	POSTED	4.3.5.1, p. 62
ALU_MCR2	2901	ALU Master Control, Field 2	POSTED	4.3.5.1, p. 62
ALU_TOP	2902	Tag Operation	POSTED	4.3.5.2, p. 64
ALU_AV	2903	Alpha Value	POSTED	4.3.5.3, p. 64
ALU_LOPY	2904	Logic Operation Channel Y	POSTED	4.3.5.4, p. 65
ALU_LOPU	2905	Logic Operation Channel U	POSTED	4.3.5.4, p. 65
ALU_LOPV	2906	Logic Operation Channel V	POSTED	4.3.5.4, p. 65
ALU_CAY	2907	Constant A, Channel Y	POSTED	4.3.5.5, p. 65
ALU_CAU	2908	Constant A, Channel U	POSTED	4.3.5.5, p. 65
ALU_CAV	2909	Constant A, Channel V	POSTED	4.3.5.5, p. 65
ALU_CBY	290a	Constant B, Channel Y	POSTED	4.3.5.6, p. 66
ALU_CBU	290b	Constant B, Channel U	POSTED	4.3.5.6, p. 66
ALU_CBV	290c	Constant B, Channel V	POSTED	4.3.5.6, p. 66
ALU_CCY	290d	Constant C, Channel Y	POSTED	4.3.5.7, p. 66
ALU_CCU	290e	Constant C, Channel U	POSTED	4.3.5.7, p. 66
ALU_CCV	290f	Constant C, Channel V	POSTED	4.3.5.7, p. 66

ATI019068



### 4.3 VPU: Video Processor Unit (cont.)

Name	Index	Definition	Posted?	Ref. Section
OPU: Output Processing Unit				4.3.6, p. 67
OPU_MCR1	2a00	OPU Master Control, Field 1	POSTED	4.3.6.1, p. 67
OPU_XBI1	2a02	X Begin Integer, Field 1	POSTED	4.3.6.2, p. 68
OPU_XEI1	2a03	X End Integer, Field 1	POSTED	4.3.6.3, p. 68
OPU_YBI1	2a07	Y Begin Integer, Field 1	POSTED	4.3.6.4, p. 69
OPU_YEI1	2a08	Y End Integer, Field 1	POSTED	4.3.6.5, p. 69
OPU_MCR2	2b00	OPU Master Control, Field 2	POSTED	4.3.6.1, p. 67
OPU_XBI2	2b02	X Begin Integer, Field 2	POSTED	4.3.6.2, p. 68
OPU_XEI2	2b03	X End Integer, Field 2	POSTED	4.3.6.3, p. 68
OPU_YBI2	2b07	Y Begin Integer, Field 2	POSTED	4.3.6.4, p. 69
OPU_YEI2	2b08	Y End Integer, Field 2	POSTED	4.3.6.5, p. 69



### 4.3.1 VPU Global Control

#### 4.3.1.1 VPU\_MCR: VPU Master Control

POSTED

I/O Address HIU\_RDT

Index 2000

Register VPU\_MCR controls the operation of the IPU1, the IPU2, and the OPU for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			ALUE	OPFSS			IP2FSS			IP1FSS					

Bit #	Access	Reset	Description
15:13	R/W	0h	RSVD Reserved (read as '0').
12	R/W	0	ALUE ALU Enable. 0 Disable ALU operation 1 Enable ALU operation
11:8	R/W	0000	OPFSS OPU Field Sync Select. Enables OPU operation, specifies field synchronization and processing. 0000 Disable OPU operation 0001 Start OPU on next field, both fields processed 0010 Start OPU on field 1, single field processed 0011 Start OPU on field 1, both fields processed 0100 Start OPU on field 2, single field processed 0101 Start OPU on field 2, both fields processed
7:4	R/W	0000	IP2FSS IPU2 Field Sync Select. Enables IPU2 operation, specifies field synchronization and processing. 0000 Disable IPU2 operation 0001 Start IPU2 on next field, both fields processed 0010 Start IPU2 on field 1, single field processed 0011 Start IPU2 on field 1, both fields processed 0100 Start IPU2 on field 2, single field processed 0101 Start IPU2 on field 2, both fields processed
3:0	R/W	0000	IP1FSS IPU1 Field Sync Select. Enables IPU1 operation, specifies field synchronization and processing. 0000 Disable IPU1 operation 0001 Start IPU1 on next field, both fields processed 0010 Start IPU1 on field 1, single field processed 0011 Start IPU1 on field 1, both fields processed 0100 Start IPU1 on field 2, single field processed 0101 Start IPU1 on field 2, both fields processed

ATI019070



#### 4.3.2 IPU1: Input Processor Unit 1

##### 4.3.2.1 IPU1\_PIX: Pixel Count

I/O Address HIU\_RDT  
Index 2100

Register IPU1\_PIX is a read-only register that reads back the value of the current 11-bit pixel counter.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						PC									

Bit #	Access	Reset	Description
15:11	R	0h	RSVD Reserved (read as '0').
10:0	R	0h	PC Pixel Count current line. Automatically resets to '0' at the beginning of each line. (0-7FFh)

##### 4.3.2.2 IPU1\_LIC: Line Count

I/O Address HIU\_RDT  
Index 2101

Register IPU1\_LIC is a read-only register of the current 11-bit line count.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						LC									

Bit #	Access	Reset	Description
15:11	R	0h	RSVD Reserved (read as '0').
10:0	R	0h	LC Line Count current field. Automatically resets to '0' at the beginning of each field. (0-7FFh)





#### 4.3.2.3 IPU1\_FLC: Field Count

I/O Address HIU\_RDT  
Index 2102

Register IPU1\_FLC returns the current 15-bit field count on read.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	FC														

Bit #	Access	Reset	Description
15	R	0h	RSVD Reserved (read as '0').
14:0	R	0h	FC Field Count. Resets to '0' when IPU1_FIR, bit FCE = '0.'

#### 4.3.2.4 IPU1\_LIR: Line Count Interrupt Request

I/O Address HIU\_RDT  
Index 2103

Register IPU1\_LIR generates an interrupt request when the 11-bit value in field IRLC is equal to the value in IPU1\_LIC, bit LC.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					IRLC										

Bit #	Access	Reset	Description
15:11	R/W	0h	RSVD Reserved (read as '0').
10:0	R/W	0h	IRLC Interrupt Request Line Count (0-7FFh)

#### 4.3.2.5 IPU1\_FIR: Field Count Interrupt Request

I/O Address HIU\_RDT  
Index 2104

Register IPU1\_FIR generates an interrupt request when the 15-bit value in field IRFC is equal to the value in IPU1\_FLC, field FC.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FCE	IRFC														

ATI019072



Bit #	Access	Reset	Description
5	R/W	0h	FCE Field Count Enable. 0 Disable field count 1 Enable field count
14 0	R/W	0h	IRFC Interrupt Request Field Count.

#### 4.3.2.6 IPU1\_LRB: LUT RAM Base Address

I/O Address HIU\_RDT  
Index 2200

Register IPU1\_LRB preloads the 8-bit LUT RAM address counter and initializes the channel pointer to the YR channel. The channel pointer automatically advances to the next channel after each LUT RAM access, and address counter automatically increments after accessing the CrB channel. LUT RAM elements are accessed in the following order: YR[LRB+0], CbG[LRB+0], CrB[LRB+0], YR[LRB+1], CbG[LRB+1], CrB[LRB+1], etc.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								LRB							

Bit #	Access	Reset	Description
15.8	R/W	0h	RSVD Reserved (read as '0').
7 0	R/W	0h	LRB LUT RAM Base Address. Specifies the 8-bit address generator preload value. (0-FFh)

#### 4.3.2.7 IPU1\_LRD: LUT RAM Data

I/O Address HIU\_RDT  
Index 2201

Register IPU1\_LRD is the bidirectional data port to the storage elements of the three-channel LUT RAM.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								LRD							

Bit #	Access	Reset	Description
15.8	R/W	0h	RSVD Reserved (read as '0').
7 0	R/W	0h	LRD LUT RAM Data. Data written to this field transfers to the current LUT RAM element (R, G, B); data to be read from the current LUT RAM element appears in this field. (0-FFh)



#### 4.3.2.8 IPU1\_MCR1: IPU1 Master Control

POSTED

I/O Address HIU\_RDT

Index 3000 (IPU1\_MCR1: IPU1 Master Control, Field 1)

3100 (IPU1\_MCR2: IPU1 Master Control, Field 2)

Registers IPU1\_MCR1 and IPU1\_MCR2 control the operation of the IPU1 for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPS	IM	PSE	CSCE	LE	YSP	OOT	OF				IF				

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15	R/W	0	FPS	Field Polarity Select. Controls the polarity of the field ID signal supplied to the Window Clipping and XY Scaler. 0 Normal polarity 1 Invert polarity
----	-----	---	-----	--

14	R/W	0	IM	Interlace Mode. Specifies the input stream as interlaced or progressive-scan (non-interlaced) data. 0 Progressive-scan input 1 Interlaced input
----	-----	---	----	---

13	R/W	0	PSE	X Prescaler Enable. 0 Bypass prescaler 1 Enable 0.5x prescaler
----	-----	---	-----	--

12	R/W	0	CSCE	Color Space Converter Enable. 0 Bypass Color Space Converter 1 Enable Color Space Converter
----	-----	---	------	---

11	R/W	0	LE	LUT Enable. 0 Bypass LUT RAM 1 Enable LUT RAM
----	-----	---	----	---

10	R/W	0	YSP	Y Scaling Path. Enables or disables the special Y Scaling Path Mode. 0 Enable IPU1 Y scaling 1 Disable IPU1 Y scaling (ALU performs Y scaling)
----	-----	---	-----	--

9:8	R/W	00	ODT	Output Data Tag. Controls input selection of Input Tag Unit tag mux. 00 Pass tag unchanged 01 Set tag to field ID 10 Set tag to inverse chroma key tag 11 Set tag to chroma key tag
-----	-----	----	-----	---

7:4	R/W	0000	OF	Output Data Stream Format. 0000 YCbCr 4:2:2 non-tagged data 0001 YCbCr 4:2:2 tagged data 1000 RGB 5:6:5 non-tagged data 1001 RGB 1:5:5:5 tagged data 1010 RGB 8:8:8 non-tagged data 1011 RGB 1:8:8:8 tagged data 1110 RGB 3:3:2 non-tagged data XXXX All other configurations reserved
-----	-----	------	----	--

ATI019074



Bit #	Access	Reset	Description
3:0	R/W	0000	IF Input Data Stream Format.
		0000	YCbCr 4:2:2 non-tagged data
		0001	YCbCr 4:2:2 tagged data
		0010	YCbCr 4:1:1 non-tagged data
		1000	RGB 5:6:5 non-tagged data
		1001	RGB 1:5:5:5 tagged data
		1110	Pseudo color (indirect color mapping via IPU1 LUT)
		XXXX	All other configurations reserved

#### 4.3.2.9 IPU1\_XBnf: X Begin

POSTED

I/O Address HIU\_RDT

Index 3001 (IPU1\_XBF1: X Begin Fraction, Field 1) 3101 (IPU1\_XBF2: X Begin Fraction, Field 2)  
3002 (IPU1\_XBI1: X Begin Integer, Field 1) 3102 (IPU1\_XBI2: X End Integer, Field 2)

Registers IPU1\_XBnf specify the 11.3 format X begin value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BF			RSVD												
RSVD						BI									

Bit #	Access	Reset	Description
-------	--------	-------	-------------

#### IPU1\_XBFf: X Begin Fraction Index

15:13	R/W	0h	BF	Begin X Column Fractional Index. Specifies the 3-bit fractional portion of the 11.3 format X begin value. Allows the virtual left boundary of the post-scaled window to be aligned between pixels of the pre-scaled window for fields 1 and 2. (0-7h)
12:0	R/W	0h	RSVD	Reserved (read as '0').

#### IPU1\_XEIf: X Begin Integer Index

15:11	R/W	0h	RSVD	Reserved (read as '0').
10:0	R/W	0h	BI	Begin X Column Integer Index. Specifies the 11-bit integer portion of the 11.3 format X begin value. Defines the left boundary of the pre-scaling window for fields 1 and 2. All video to the left of this boundary is clipped and is not used to generate the scaled window. (0-7FFh)

ATI019075



#### 4.3.2.10 IPU1\_XEI: X End

POSTED

I/O Address HIU\_RDT

Index 3003 (IPU1\_XEI1: X End Integer, Field 1)

3103 (IPU1\_XEI2: X End Integer, Field 2)

Registers IPU1\_XEI1 and IPU1\_XEI2 specify the 11-bit X end value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						EI									

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:11	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

10:0	R/W	0h	EI X End Column Integer Index. Specifies the 11-bit X end value. (0-7FFh)
------	-----	----	---

#### 4.3.2.11 IPU1\_XSnf: X Shrink

POSTED

I/O Address HIU\_RDT

Index 3004 (IPU1\_XSF1: X Shrink Fraction, Field 1)

3005 (IPU1\_XSI1: X Shrink Integer, Field 1)

3104 (IPU1\_XSF2: X Shrink Fraction, Field 2)

3105 (IPU1\_XSI2: X Shrink Integer, Field 2)

Registers IPU1\_XSnf specify the 6.10 format X shrink value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SF										RSVD					
RSVD										SI					

Bit #	Access	Reset	Description
-------	--------	-------	-------------

##### IPU1\_XSF1: X Shrink Fraction

15:5	R/W	0h	SF X Shrink Fraction. Specifies the 10-bit fractional portion of the 6.10 format X shrink value. (0-3FFh)
------	-----	----	---

4:0	R/W	0h	RSVD Reserved (read as '0').
-----	-----	----	------------------------------

##### IPU1\_XSI1: X Shrink Integer

15:6	R/W	0h	RSVD Reserved (read as '0').
------	-----	----	------------------------------

5:0	R/W	0h	SI X Shrink Integer. Specifies the 4-bit integer portion of the 4.10 format X shrink value. (0-Fh)
-----	-----	----	--

ATI019076



#### 4.3.2.12 IPU1\_YBnf: Y Begin

POSTED

I/O Address HIU\_RDT

Index 3006 (IPU1\_YBF1: Y Begin Fraction, Field 1) 3106 (IPU1\_YBF2: Y Begin Fraction, Field 2)  
3007 (IPU1\_YBI1: Y Begin Integer, Field 1) 3107 (IPU1\_YBI2: Y Begin Integer, Field 2)

Registers IPU1\_YBnf specify the 11.3 format Y begin value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BF			RSVD												
RSVD						BI									

Bit #	Access	Reset	Description
-------	--------	-------	-------------

##### IPU1\_YBFf: Y Begin Fraction Index

15:13	R/W	0h	BF	Begin Y Row Fractional Index. Specifies the 3-bit fractional portion of the 11.3 format Y begin value. Allows the virtual top row of the post-scaled window to be aligned between rows of the pre-scaled window for fields 1 and 2. (0-7h)
-------	-----	----	----	--

12:0	R/W	0h	RSVD	Reserved (read as '0').
------	-----	----	------	-------------------------

##### IPU1\_YBIf: Y Begin Integer Index

15:11	R/W	0h	RSVD	Reserved (read as '0').
-------	-----	----	------	-------------------------

10:0	R/W	0h	BI	Begin Y Row Integer Index. Specifies the 11-bit integer portion of the 11.3 format Y begin value. Defines the top edge of the pre-scaling window for fields 1 and 2. All video above this boundary is clipped and does not become part of the scaled window. (0-7FFh)
------	-----	----	----	---

#### 4.3.2.13 IPU1\_YEIf: Y End

POSTED

I/O Address HIU\_RDT

Index 3008 (IPU1\_YEI1: Y End Integer, Field 1) 3108 (IPU1\_YEI2: Y End Integer, Field 2)

Registers IPU1\_YEI1 and IPU1\_YEI2 specify the 11-bit Y end value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						EI									

Bit #	Access	Reset	Description
-------	--------	-------	-------------

ATI019077 -

15:11	R/W	0h	RSVD	Reserved (read as '0').
-------	-----	----	------	-------------------------

10:0	R/W	0h	EI	End Y Row Integer Index. Specifies the 11-bit Y end value. (0-7FFh)
------	-----	----	----	---



#### 4.3.2.14 IPU1\_YSnf: Y Shrink

POSTED

I/O Address HIU\_RDT

Index 3009 (IPU1\_YSF1: Y Shrink Fraction, Field 1) 3109 (IPU1\_YSF2: Y Shrink Fraction, Field 2)  
300a (IPU1\_YSI1: Y Shrink Integer, Field 1) 310a (IPU1\_YSI2: Y Shrink Integer, Field 2)

Registers IPU1\_YSnf specify the 4.10 format Y shrink value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SF										RSVD					
RSVD										SI					

Bit #	Access	Reset	Description
-------	--------	-------	-------------

##### IPU1\_YSF1: Y Shrink Fraction

15:8	R/W	0h	SF	Y Shrink Fraction. Specifies the 10-bit fractional portion of the 4.10 format Y shrink value. (0-3FFh)
5:0	R/W	0h	RSVD	Reserved (read as '0').

##### IPU1\_YSI1: Y Shrink Integer

15:6	R/W	0h	RSVD	Reserved (read as '0').
5:0	R/W	0h	SI	Y Shrink Integer. Specifies the 4-bit integer portion of the 4.10 format Y shrink value. (0-Fh)

#### 4.3.2.15 IPU1\_KFC1: Key Function Code

POSTED

I/O Address HIU\_RDT

Index 300b (IPU1\_KFC1: Key Function Code, Field 1) 310b (IPU1\_KFC2: Key Function Code, Field 2)

Registers IPU1\_KFC1 and IPU1\_KFC2 specify the tag values used by the key function code multiplexers for fields 1 and 2 in the tag unit, allowing a match on any combination of YUV to trigger a tag.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								KEYFC							

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:8	R/W	0h	RSVD	Reserved (read as '0').												
7:0	R/W	0h	KEYFC	Key Function Code. Specifies eight 1-bit input tag values used by the key function code multiplexers. (1 = match, or within range.) (0-FFh)												
				<table><tr><td>YUV</td><td>YUV</td><td>YUV</td><td>YUV</td></tr><tr><td>000 = KEYFC0</td><td>010 = KEYFC2</td><td>100 = KEYFC4</td><td>110 = KEYFC6</td></tr><tr><td>001 = KEYFC1</td><td>011 = KEYFC3</td><td>101 = KEYFC5</td><td>111 = KEYFC7</td></tr></table>	YUV	YUV	YUV	YUV	000 = KEYFC0	010 = KEYFC2	100 = KEYFC4	110 = KEYFC6	001 = KEYFC1	011 = KEYFC3	101 = KEYFC5	111 = KEYFC7
YUV	YUV	YUV	YUV													
000 = KEYFC0	010 = KEYFC2	100 = KEYFC4	110 = KEYFC6													
001 = KEYFC1	011 = KEYFC3	101 = KEYFC5	111 = KEYFC7													



#### 4.3.2.16 IPU1\_MMxf: Chroma Key Max/Min

POSTED

I/O Address	HIU_RDT
Index	300c (IPU1_MMY1: Chroma Key Y/R Max/Min, Field 1)
	300d (IPU1_MMU1: Chroma Key U/G Max/Min, Field 1)
	300e (IPU1_MMV1: Chroma Key V/B Max/Min, Field 1)
	310c (IPU1_MMY2: Chroma Key Y/R Max/Min, Field 2)
	310d (IPU1_MMU2: Chroma Key U/G Max/Min, Field 2)
	310e (IPU1_MMV2: Chroma Key V/B Max/Min, Field 2)

Registers IPU1\_MMxf specify the maximum and minimum 8-bit chroma key comparator values used by the Input Tag Unit for fields 1 and 2. These values are used for each of three 8-bit input channels for both fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
YRMAX								YRMIN							
UGMAX								UGMIN							
VBMAX								VBMIN							

Bit # Access Reset Description

##### IPU1\_MMYf: Key Y/R Maximum/Minimum

15:8	R/W	0h	YRMAX	Key Y/R Maximum. Specifies the upper threshold for the 8-bit Y/R channel comparator. (0-FFh)
7:0	R/W	0h	YRMIN	Key Y/R Minimum. Specifies the lower threshold for the 8-bit Y/R channel comparator. (0-FFh)

##### IPU1\_MMUf: Key U/G Maximum/Minimum

15:8	R/W	0h	UGMAX	Key U/G Maximum. Specifies the upper threshold for the 8-bit Cb/G channel comparator. (0-FFh)
7:0	R/W	0h	UGMIN	Key U/G Minimum. Specifies the lower threshold for the 8-bit Cb/G channel comparator. (0-FFh)

##### IPU1\_MMVf: Key V/B Maximum/Minimum

15:8	R/W	0h	VBMAX	Key V/B Maximum. Specifies the upper threshold for the 8-bit Cr/B channel comparator. (0-FFh)
7:0	R/W	0h	VBMIN	Key V/B Minimum. Specifies the lower threshold for the 8-bit Cr/B channel comparator. (0-FFh)

ATI019079



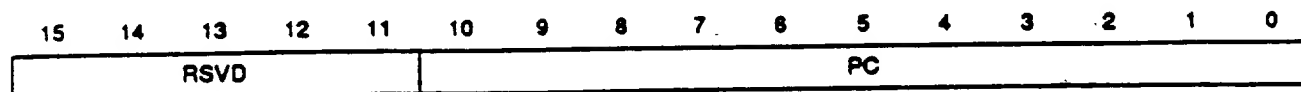


### 4.3.3 IPU2: Input Processing Unit 2

#### 4.3.3.1 IPU2\_PIX: Pixel Count

I/O Address HIU\_RDT  
Index 2300

Register IPU2\_PIX is a read-only register that specifies the current 11-bit pixel count.

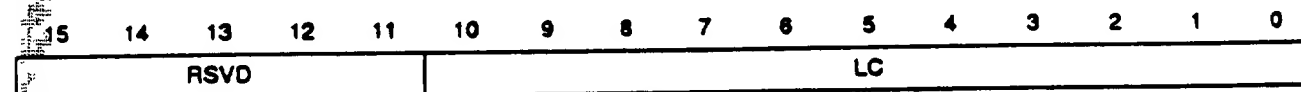


Bit #	Access	Reset	Description
15:11	R	0h	RSVD Reserved (read as '0').
10:0	R	0h	PC Pixel Count current line. Automatically resets to '0' at the beginning of each line. (0-7FFh)

#### 4.3.3.2 IPU2\_LIC: Line Count

I/O Address HIU\_RDT  
Index 2301

Register IPU2\_LIC is a read-only register that specifies the current 11-bit line count.

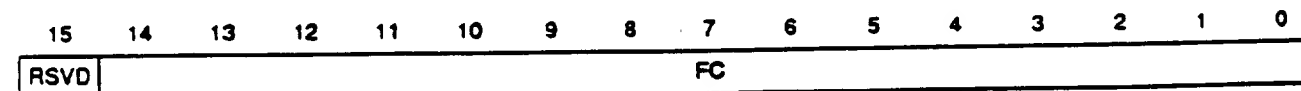


Bit #	Access	Reset	Description
15:11	R	0h	RSVD Reserved (read as '0').
10:0	R	0h	LC Line Count current field. Automatically resets to '0' at the beginning of each field. (0-7FFh)

#### 4.3.3.3 IPU2\_FLC: Field Count

I/O Address HIU\_RDT  
Index 2302

Read-only register IPU2\_FLC returns the current 15-bit field count.



Bit #	Access	Reset	Description
15	R	0h	RSVD Reserved (read as '0').
14:0	R	0h	FC Field count.

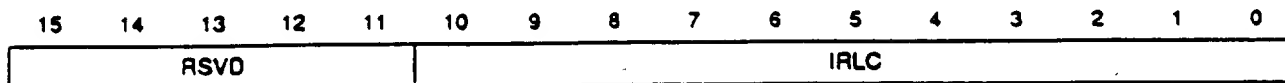
ATI019080



#### 4.3.3.4 IPU2\_LIR: Line Count Interrupt Request

I/O Address HIU\_RDT  
Index 2303

Register IPU2\_LIR specifies an 11-bit line count value that generates an interrupt request when equal to the realtime line count value in register IPU2\_LIC.

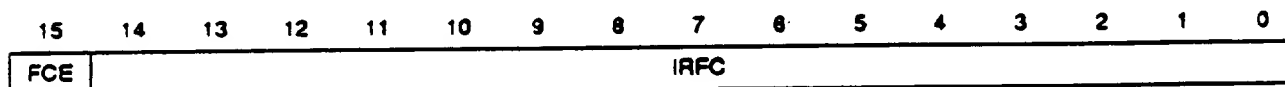


Bit #	Access	Reset	Description
15:11	R/W	0h	RSVD Reserved (read as '0').
10:0	R/W	0h	IRLC Interrupt Request Line Count. (0-7FFh)

#### 4.3.3.5 IPU2\_FIR: Field Count Interrupt Request

I/O Address HIU\_RDT  
Index 2304

Register IPU2\_FIR specifies a 16-bit field count value that generates an interrupt request when equal to the realtime field count value in register IPU2\_FLC.



Bit #	Access	Reset	Description
15	R/W	0	FCE Field Count Enable. 1 Enable field count 0 Disable field count
14:0	R/W	0h	IRFC Interrupt Request Field Count.

ATI019081



#### 4.3.3.6 IPU2\_MCR1: IPU2 Master Control

POSTED

I/O Address HIU\_RDT

Index 3200 (IPU2\_MCR1: IPU2 Master Control, Field 1)

3300 (IPU2\_MCR2: IPU2 Master Control, Field 2)

Registers IPU2\_MCR1 and IPU2\_MCR2 control the operation of the IPU2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPS	IM	PSE	RSVD												

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15	R/W	0h	FPS	Field Polarity Select. Controls the polarity of the field ID signal supplied to the XY Window Clipping subunit. 0 Normal polarity 1 Invert polarity
14	R/W	0h	IM	Interface Mode. Specifies the input stream as interlaced or non-interlaced (progressive-scan) data. 0 Progressive-scan input 1 Interlaced input
13	R/W	0h	PSE	Prescaler Enable. Enables or disables the operation of the Prescaler. 0 Bypass Prescaler 1 Enable 0.5x Prescaler
12:0	R/W	0h	RSVD	Reserved (read as '0').

#### 4.3.3.7 IPU2\_XBI1: X Begin

POSTED

I/O Address HIU\_RDT

Index 3202 (IPU2\_XBI1: X Begin Integer, Field 1)

3302 (IPU2\_XBI2: X Begin Integer, Field 2)

Registers IPU2\_XBI1 and IPU2\_XBI2 specify the 11-bit X begin value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											BI				

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:11	R/W	0h	RSVD	Reserved (read as '0').
10:0	R/W	0h	BI	Begin X Column Integer Index. Specifies the 11-bit integer portion of the 11.0 format X begin value. (0-7FFh)

ATI019082



#### 4.3.3.8 IPU2\_XEI1: X End

POSTED

I/O Address HIU\_RDT

Index 3203 (IPU2\_XEI1: X End Integer, Field 1)

3303 (IPU2\_XEI2: X End Integer, Field 2)

Registers IPU2\_XEI1 and IPU2\_XEI2 specify the 11-bit X end value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						EI									

Bit #	Access	Reset	Description
15:11	R/W	0h	RSVD Reserved (read as '0').
10:0	R/W	0h	EI End X Column Integer Index. Specifies the 11-bit X end value. (0-7FFh)

#### 4.3.3.9 IPU2\_YBI1: Y Begin

POSTED

I/O Address HIU\_RDT

Index 3207 (IPU2\_YBI1: Y Begin Integer, Field 1)

3307 (IPU2\_YBI2: Y Begin Integer, Field 2)

Registers IPU2\_YBI1 and IPU2\_YBI2 specify the 11-bit Y begin value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						BI									

Bit #	Access	Reset	Description
15:11	R/W	0h	RSVD Reserved (read as '0').
10:0	R/W	0h	BI Begin Y Row Integer Index. Specifies the 11-bit integer portion of the 11-bit Y begin value. (0-7FFh)

ATI019083



#### 4.3.3.10 IPU2\_YEI: Y End

POSTED

I/O Address HIU\_RDT

Index 3208 (IPU2\_YEI1: Y End Integer, Field 1)  
3308 (IPU2\_YEI2: Y End Integer, Field 2)

Registers IPU2\_YEI1 and IPU2\_YEI2 specify the 11-bit Y end value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						EI									

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:11	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

10:0	R/W	0h	EI End Y Row Integer Index. Specifies the 11-bit Y end value. (0-7FFh)
------	-----	----	--

#### 4.3.4 SIU: Sequencer Instruction Unit

##### 4.3.4.1 SIU\_MCR: SIU Master Control

I/O Address HIU\_RDT

Index 2800

Register SIU\_MCR controls the operation of the SIU for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		SE		FT		SI2					SI1				

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:14	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

13:12	R/W	00	SE Sequencer Enable. Enables the operation of the SIU.
-------	-----	----	--

00 SIU disabled

10 SIU enabled, start on SI1

11 SIU enabled, start on SI2

11:10	R/W	00	FT Field Toggle. Specifies the field toggle mode and the association of the start index values to a field.
-------	-----	----	--

00 No field toggle

01 SI1 and SI2 toggle on vertical sync; no field association

10 Field 1 associated to SI1, fields 1 and 2 toggle on vertical sync

11 Field 1 associated to SI2, fields 1 and 2 toggle on vertical sync

9:5	R/W	0h	SI2 Start Index 2. Specifies the 5-bit sequencer instruction Start Index 2. (0-1Fh)
-----	-----	----	---

4:0	R/W	0h	SI1 Start Index 1. Specifies the 5-bit sequencer instruction Start Index 1. (0-1Fh)
-----	-----	----	---

AT1019084



#### 4.3.4.2 SIU\_FCS: FIFO Control/Status

IO Address HIU\_RDT  
Index 2801

Register SIU\_FCS is a special read/write register that provides realtime access to the full and empty flags from FIFOs A-G. All flags are active high.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	FGF	FGE	FFF	FFE	FEF	FEE	FDF	FDE	FCF	FCE	FBF	FBE	FAF	FAE	

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:14	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

13	R	0h	FGF FIFO G Full Flag
----	---	----	----------------------

12	R/W	0h	FGE FIFO G Empty Flag
----	-----	----	-----------------------

11	R	0h	FFF FIFO F Full Flag
----	---	----	----------------------

10	R/W	0h	FFE FIFO F Empty Flag
----	-----	----	-----------------------

9	R	0h	FEF FIFO E Full Flag
---	---	----	----------------------

8	R/W	0h	FEE FIFO E Empty Flag
---	-----	----	-----------------------

7	R	0h	FDF FIFO D Full Flag
---	---	----	----------------------

6	R/W	0h	FDE FIFO D Empty Flag
---	-----	----	-----------------------

5	R	0h	FCF FIFO C Full Flag
---	---	----	----------------------

4	R/W	0h	FCE FIFO C Empty Flag
---	-----	----	-----------------------

3	R	0h	FBF FIFO B Full Flag
---	---	----	----------------------

2	R/W	0h	FBE FIFO B Empty Flag
---	-----	----	-----------------------

1	R	0h	FAF FIFO A Full Flag
---	---	----	----------------------

0	R/W	0h	FAE FIFO A Empty Flag
---	-----	----	-----------------------

For all FIFO Full flags (odd bits 13:1):  
0 Enables FIFO  
1 Resets FIFO

For all FIFO Empty flags (even bits 12:0):  
1 FIFO is empty

AT1019007



#### 4.3.4.3 SIU\_FOU: FIFO Overflow/Underflow

I/O Address HIU\_RDT  
Index 2802

Register SIU\_FOU is a read-only register that provides realtime access to the overflow and underflow flags from FIFOs A-G.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD	FGO	FGU	FFO	FFU	FEO	FEU	FDO	FDU	FCO	FCU	FBO	FBU	FAO	FAU	

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:14	R	0h	RSVD	Reserved (read as '0').
13	R	0h	FGO	FIFO G Overflow Flag
12	R	0h	FGU	FIFO G Underflow Flag
11	R	0h	FFO	FIFO F Overflow Flag
10	R	0h	FFU	FIFO F Underflow Flag
9	R	0h	FEO	FIFO E Overflow Flag
8	R	0h	FEU	FIFO E Underflow Flag
7	R	0h	FDO	FIFO D Overflow Flag
6	R	0h	FDU	FIFO D Underflow Flag
5	R	0h	FCO	FIFO C Overflow Flag
4	R	0h	FCU	FIFO C Underflow Flag
3	R	0h	FBO	FIFO B Overflow Flag
2	R	0h	FBU	FIFO B Underflow Flag
1	R	0h	FAO	FIFO A Overflow Flag
0	R	0h	FAU	FIFO A Underflow Flag

For all FIFO Overflow flags  
(odd bits 13:1):

0 Resets FIFO  
1 FIFO overflow

For all FIFO Underflow flags  
(even bits 12:0):

0 Resets FIFO  
1 FIFO underflow

ATI019086



#### 4.3.4.4 SIUs\_SIM: Sequencer Instruction Memory

I/O Address	HIU_RDT			
Index	2e00 (SIU0_SIM)	2e08 (SIU8_SIM)	2e10 (SIU16_SIM)	2e18 (SIU24_SIM)
	2e01 (SIU1_SIM)	2e09 (SIU9_SIM)	2e11 (SIU17_SIM)	2e19 (SIU25_SIM)
	2e02 (SIU2_SIM)	2e0a (SIU10_SIM)	2e12 (SIU18_SIM)	2e1a (SIU26_SIM)
	2e03 (SIU3_SIM)	2e0b (SIU11_SIM)	2e13 (SIU19_SIM)	2e1b (SIU27_SIM)
	2e04 (SIU4_SIM)	2e0c (SIU12_SIM)	2e14 (SIU20_SIM)	2e1c (SIU28_SIM)
	2e05 (SIU5_SIM)	2e0d (SIU13_SIM)	2e15 (SIU21_SIM)	2e1d (SIU29_SIM)
	2e06 (SIU6_SIM)	2e0e (SIU14_SIM)	2e16 (SIU22_SIM)	2e1e (SIU30_SIM)
	2e07 (SIU7_SIM)	2e0f (SIU15_SIM)	2e17 (SIU23_SIM)	2e1f (SIU31_SIM)

The 32 identical registers SIUs\_SIM store the instruction sequence for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD		OTN				EP		FA				OBA			

Bit #	Access	Reset	Description
15:14	R/W	0h	RSVD Reserved (read as '0').
13:9	R/W	0h	OTN Offset to Next Instruction. Specifies the signed, 5-bit displacement to the next instruction to execute. (0-1Fh)
8	R/W	0	EP Exit Point. Identifies the current instruction as the exit point when the field toggle condition is detected. 0 Normal fall-through instruction 1 Exit point instruction
7:4	R/W	0000	FA FIFO Association. Associates a FIFO with the current instruction. 0000 FIFO G 0001 FIFO F 0010 FIFO E 0011 FIFO A 0100 FIFO B 0101 FIFO C 0110 FIFO D XXXX All other configurations reserved
3:0	R/W	0000	OBA Object Buffer Association. Associates an object buffer with the current instruction (see field FA). 0000 Object buffer 0 0001 Object buffer 1 0010 Object buffer 2 0011 Object buffer 3 0100 Object buffer 4 0101 Object buffer 5 0110 Object buffer 6 0111 Object buffer 7

ATI019087





#### 4.3.4.5 SIU\_FAR: FIFO Auto Reset

I/O Address HIU\_RDT  
Index 4001

Register SIU\_FAR controls whether SIU FIFOs F and G are automatically reset (cleared) or the vertical sync from the video source designated as master in register VIU\_WDT, bit MFTS.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								FGR	FFR	RSVD					

Bit #	Access	Reset	Description
15:7	R/W	0h	RSVD Reserved (read as '0').
6	R/W	0	FGR FIFO G Reset. Reset FIFO G on the vertical sync master in register VIU_WDT, bit MFTS. 0 no auto reset 1 clear FIFO G on vertical sync.
5	R/W	0	FFR FIFO F Reset. Reset FIFO F on the vertical sync master in register VIU_WDT, bit MFTS. 0 no auto reset 1 clear FIFO F on vertical sync.
4:0	R/W	0h	RSVD Reserved (read as '0').

#### 4.3.5 ALU: Arithmetic and Logic Unit

##### 4.3.5.1 ALU\_MCR1: ALU Master Control

I/O Address HIU\_RDT  
Index 2900 (ALU\_MCR1: ALU Master Control, Field 1)  
2901 (ALU\_MCR2: ALU Master Control, Field 2)

Registers ALU\_MCR1 and ALU\_MCR2 specify the ALU operating mode for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GBM	TF		AOP			YOUT		UOUT		VOUT		OPCS	OPBS	OPAS	

Bit #	Access	Reset	Description
15	R/W	00	GBM Three-operand Bit Mask selecting tag source. 0 Bit per bit mask — bit n in FIFO C will mask bit n of the pixel currently operated on (from FIFO A or B) by the ALU 1 Bit per pixel mask — bit n in FIFO C will mask pixel n from FIFO A or B

ATI019088



Bit #	Access	Reset	Description
14:13	R/W	0000	TF Tag Format. Specifies both the input and output stream tag formats. 00 No tag 01 Tagged 4:2:2 YCbCr data 10 Tagged 5:5:5 RGB data 11 Tagged 8:8:8 RGB data
12:9	R/W	00	AOP Arithmetic Operation Select. 0000 Alpha mix using alpha register ( $dA + (1-d)B$ ) 0001 Alpha mix using operand C ( $cA + (1-c)B$ ) 0010 Operand A + Operand B 0011 Operand A - Operand B 0100 (Operand A - Operand B) / 2 0101 Reconstruct field from operands A and B 0110 Four frame interpolate from operands A and B XXXX All other configurations are reserved; results are unpredictable
8:7	R/W	00	YOUT Y Output Source Select. 00 Source output from logical unit 01 Source output from arithmetic unit 10 Source output based on control tag 11 Enable arithmetic out based on tag
6:5	R/W	00	UOUT U Output Source Select. 00 Source output from logical unit 01 Source output from arithmetic unit 10 Source output based on control tag 11 Enable arithmetic out based on tag
4:3	R/W	0	VOUT V output Source Select. 00 Source output from logical unit 01 Source output from arithmetic unit 10 Source output based on control tag 11 Enable arithmetic out based on tag
2	R/W	0	OPCS Operand C Source Select. 0 Operand sourced from constant register 1 Operand sourced from FIFO
1	R/W	0	OPBS Operand B Source Select. 0 Operand sourced from constant register 1 Operand sourced from FIFO
0	R/W	0	OPAS Operand A Source Select. 0 Operand sourced from constant register 1 Operand sourced from FIFO

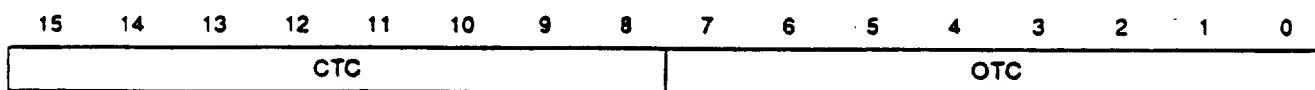
ATI019089



#### 4.3.5.2 ALU\_TOP: Tag Operation

I/O Address HIU\_RDT  
Index 2902

Register ALU\_TOP specifies the control and output tag multiplexer operation codes.

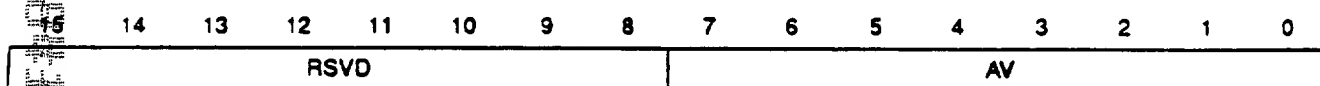


Bit #	Access	Reset	Description
15:8	R/W	0h	CTC Control Tag Code. (0-FFh)
7:0	R/W	0h	OTC Output Tag Code. (0-FFh)

#### 4.3.5.3 ALU\_AV: Alpha Value

I/O Address HIU\_RDT  
Index 2903

Register ALU\_AV specifies the alpha mix constant.



Bit #	Access	Reset	Description
15:8	R/W	0h	RSVD Reserved (read as '0').
7:0	R/W	0h	AV Alpha Value. (0-FFh)

ATI019090



#### 4.3.5.4 ALU\_LOPx: Logic Operation

I/O Address HIU\_RDT  
Index 2904 (ALU\_LOPY: Logic Operation Channel Y)  
2905 (ALU\_LOPU: Logic Operation Channel U)  
2906 (ALU\_LOPV: Logic Operation Channel V)

Registers ALU\_LOPY, ALU\_LOPU, and ALU\_LOPV specify the constant values for logical multiplexers A, B, and C, respectively.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MLOP															

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:0	R/W	0h	MLOP Multiplexor Logical Operation.
------	-----	----	-------------------------------------

#### 4.3.5.5 ALU\_CAx: Constant A

I/O Address HIU\_RDT  
Index 2907 (ALU\_CAY: Constant A, Channel Y)  
2908 (ALU\_CAU: Constant A, Channel U)  
2909 (ALU\_CAV: Constant A, Channel V)

See also: ALU\_MCRf: ALU Master Control, p. 62

Registers ALU\_CAY, ALU\_CAU, and ALU\_CAV specify the constant values for Operand A, based on the value of register ALU\_MCRf, field OPAS.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								CON							
RSVD								TAG	CON						

Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:9	R/W	0h	RSVD Reserved (read as '0').
------	-----	----	------------------------------

8	R/W	0h	TAG Tag. (ALU_CAU and ALU_CAV only) Specifies the constant tag bit to insert in the input stream. (0-1h)
---	-----	----	--

7:0	R/W	0h	CON Constant. Specifies the constant 8-bit value to use in place of the real-time input stream channel for operand A. (0-FFh)
-----	-----	----	---

ATI019091



#### 4.3.5.6 ALU\_CBx: Constant B

I/O Address HIU\_RDT  
Index 290a (ALU\_CBY: Constant B, Channel Y)  
290b (ALU\_CBU: Constant B, Channel U)  
290c (ALU\_CBV: Constant B, Channel V)

See also: ALU\_MCRf: ALU Master Control, p. 62

Registers ALU\_CBY, ALU\_CBU, and ALU\_CBV specify the constant values for Operand B, based on the value of register ALU\_MCRf, field OPBS.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								CON							
RSVD								TAG	CON						

Bit #	Access	Reset	Description
15:9	R/W	0h	RSVD Reserved (read as '0').
8	R/W	0h	TAG Tag. (ALU_CBU and ALU_CBV only) Specifies the constant tag bit to insert in the input stream. (0-1h)
7:0	R/W	0h	CON Constant. Specifies the constant 8-bit value to use in place of the real-time input stream channel for operand B. (0-FFh)

#### 4.3.5.7 ALU\_CCx: Constant C

I/O Address HIU\_RDT  
Index 290d (ALU\_CCY: Constant C, Channel Y)  
290e (ALU\_CCU: Constant C, Channel U)  
290f (ALU\_CCV: Constant C, Channel V)

See also: ALU\_MCRf: ALU Master Control, p. 62

Registers ALU\_CCY, ALU\_CCU, and ALU\_CCV specify the constant values for Operand C, based on the value of register ALU\_MCRf, field OPCS.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								CON							
RSVD								TAG	CON						

Bit #	Access	Reset	Description
15:9	R/W	0h	RSVD Reserved (read as '0').
8	R/W	0h	TAG Tag. (ALU_CCU and ALU_CCV only) Specifies the constant tag bit to insert in the input stream. (0-1h)

ATI019092



Bit #	Access	Reset	Description
7:0	R/W	0h	CON Constant. Specifies the constant 8-bit value to use in place of the real-time input stream channel for operand C. (0-FFh)

### 4.3.6 OPU: Output Processing Unit

#### 4.3.6.1 OPU\_MCR1: OPU Master Control

POSTED

I/O Address HIU\_RDT

Index 2a00 (OPU\_MCR1: OPU Master Control, Field 1) 2b00 (OPU\_MCR2: OPU Master Control, Field 2)

Registers OPU\_MCR1 and OPU\_MCR2 control the operation of the OPU for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FPS	IM	ZE	RSVD	LSM	RSVD						IF				

Bit #	Access	Reset	Description
15	R/W	0	FPS Field Polarity Select. Controls the polarity of the field ID signal supplied to the OPU Window Clipping Unit. 0 Normal polarity 1 Invert polarity
14	R/W	0	IM Interface Mode. Specifies the input stream as interlaced or non-interlaced data. 0 Progressive scan input 1 Interlaced input
13	R/W	0	ZE Zoom Enable. Enables or disables the operation of the 2:1 X zoom subunit. Only affects source object size. 0 Disable zoom 1 Enable 2X zoom
12:11	R/W	0h	RSVD Reserved (read as '0').
10	R/W	0	LSM Line Start Mode. 0 Active line starts on horizontal blank inactive 1 Active line starts on horizontal sync inactive (VSU_HAD must = VSU_HSW+3 in loopback mode.)
9:4	R/W	0h	RSVD Reserved (read as '0').
3:0	R/W	0000	IF Input Data Format. Specifies the format of the input data stream. 0000 YCbCr 4:2:2 non-tagged data 0001 YCbCr 4:2:2 tagged data 1000 RGB 5:6:5 non-tagged data 1001 RGB 5:5:5 tagged data 1110 RGB 3:3:2 non-tagged data (non-zoom mode only)

ATI019093

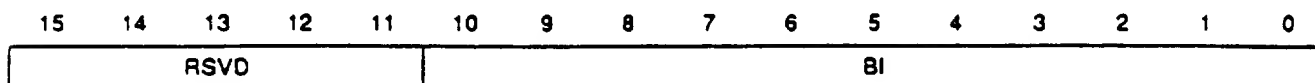


#### 4.3.6.2 OPU\_XBI1: X Begin

POSTED

I/O Address HIU\_RDT  
Index 2a02 (OPU\_XBI1: X Begin Integer, Field 1)  
2b02 (OPU\_XBI2: X Begin Integer, Field 2)

Registers OPU\_XBI1 and OPU\_XBI2 specify the 11-bit X begin value for fields 1 and 2.



Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:11	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

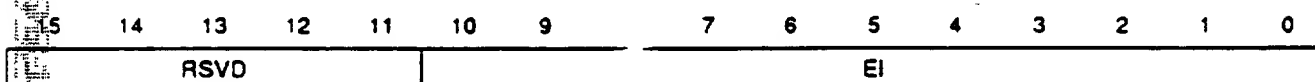
10:0	R/W	0h	BI Begin X Column Integer Index. Specifies the 11-bit integer portion of the 11-bit X begin value. (0-7FFh)
------	-----	----	---

#### 4.3.6.3 OPU\_XEI1: X End

POSTED

I/O Address HIU\_RDT  
Index 2a03 (OPU\_XEI1: X End Integer, Field 1)  
2b03 (OPU\_XEI2: X End Integer, Field 2)

Registers OPU\_XEI1 and OPU\_XEI2 specify the 11-bit X end value for fields 1 and 2.



Bit #	Access	Reset	Description
-------	--------	-------	-------------

15:11	R/W	0h	RSVD Reserved (read as '0').
-------	-----	----	------------------------------

10:0	R/W	0h	EI End X Column Integer Index. Specifies the 11-bit X end value. (0-7FFh)
------	-----	----	---

ATI019094



#### 4.3.6.4 OPU\_YBI1: Y Begin

POSTED

I/O Address HIU\_RDT

Index 2a07 (OPU\_YBI1: Y Begin Integer, Field 1)

2b07 (OPU\_YBI2: Y Begin Integer, Field 2)

Registers OPU\_YBI1 and OPU\_YBI2 specify the 11-bit Y begin value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						BI									

Bit #	Access	Reset	Description
15:11	R/W	0h	RSVD Reserved (read as '0').
10:0	R/W	0h	BI Begin Y Row Integer Index. Specifies the 11-bit integer portion of the 11-bit Y begin value. (0-7FFh)

#### 4.3.6.5 OPU\_YEI1: Y End

POSTED

I/O Address HIU\_RDT

Index 2a08 (OPU\_YEI1: Y End Integer, Field 1)

2b08 (OPU\_YEI2: Y End Integer, Field 2)

Registers OPU\_YEI1 and OPU\_YEI2 specify the 11-bit Y end value for fields 1 and 2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						EI									

Bit #	Access	Reset	Description
15:11	R/W	0h	RSVD Reserved (read as '0').
10:0	R/W	0h	EI End Y Row Integer Index. Specifies the 11-bit Y end value. (0-7FFh)

ATI019095





#### 4.3.7 RFU: Reference Frame Unit

Name	Index	Definition	Posted?	Ref. Section
<b>MMU: Memory Management Unit</b>				
MMU_MCR	4000	MMU Master Control	POSTED	4.3.2.1, p. 73
<b>OBU: Object Buffer Unit</b>				4.3.9, p. 74
OBU0_MCR	4800	Object Buffer 0 Master Control	POSTED	4.3.9.1, p. 74
OBU0_RFX	4801	Object Buffer 0 Reference Frame X Size	POSTED	4.3.9.2, p. 75
OBU0_LSL	4802	Object Buffer 0 Linear Start Address Low	POSTED	4.3.10, p. 76
OBU0_LSH	4803	Object Buffer 0 Linear Start Address High	POSTED	4.3.10, p. 76
OBU0_BSX	4804	Object Buffer 0 X Size	POSTED	4.3.10.1, p. 77
OBU0_BSY	4805	Object Buffer 0 Y Size	POSTED	4.3.10.1, p. 77
OBU0_DEC	4806	Object Buffer 0 Decimate Control	POSTED	4.3.10.2, p. 78
OBU1_MCR	4810	Object Buffer 1 Master Control	POSTED	4.3.9.1, p. 74
OBU1_RFX	4811	Object Buffer 1 Reference Frame X Size	POSTED	4.3.9.2, p. 75
OBU1_LSL	4812	Object Buffer 1 Linear Start Address Low	POSTED	4.3.10, p. 76
OBU1_LSH	4813	Object Buffer 1 Linear Start Address High	POSTED	4.3.10, p. 76
OBU1_BSX	4814	Object Buffer 1 X Size	POSTED	4.3.10.1, p. 77
OBU1_BSY	4815	Object Buffer 1 Y Size	POSTED	4.3.10.1, p. 77
OBU1_DEC	4816	Object Buffer 1 Decimate Control	POSTED	4.3.10.2, p. 78
OBU2_MCR	4820	Object Buffer 2 Master Control	POSTED	4.3.9.1, p. 74
OBU2_RFX	4821	Object Buffer 2 Reference Frame X Size	POSTED	4.3.9.2, p. 75
OBU2_LSL	4822	Object Buffer 2 Linear Start Address Low	POSTED	4.3.10, p. 76
OBU2_LSH	4823	Object Buffer 2 Linear Start Address High	POSTED	4.3.10, p. 76
OBU2_BSX	4824	Object Buffer 2 X Size	POSTED	4.3.10.1, p. 77
OBU2_BSY	4825	Object Buffer 2 Y Size	POSTED	4.3.10.1, p. 77
OBU2_DEC	4826	Object Buffer 2 Decimate Control	POSTED	4.3.10.2, p. 78
OBU3_MCR	4830	Object Buffer 3 Master Control	POSTED	4.3.9.1, p. 74
OBU3_RFX	4831	Object Buffer 3 Reference Frame X Size	POSTED	4.3.9.2, p. 75
OBU3_LSL	4832	Object Buffer 3 Linear Start Address Low	POSTED	4.3.10, p. 76
OBU3_LSH	4833	Object Buffer 3 Linear Start Address High	POSTED	4.3.10, p. 76



#### 4.3.7 RFU: Reference Frame Unit (cont.)

Name	Index	Definition	Posted?	Ref. Section
OBU3_BSX	4834	Object Buffer 3 X Size	POSTED	4.3.10.1, p. 77
OBU3_BSY	4835	Object Buffer 3 Y Size	POSTED	4.3.10.1, p. 77
OBU3_DEC	4836	Object Buffer 3 Decimate Control	POSTED	4.3.10.2, p. 78
OBU4_MCR	4840	Object Buffer 4 Master Control	POSTED	4.3.9.1, p. 74
OBU4_RFX	4841	Object Buffer 4 Reference Frame X Size	POSTED	4.3.9.2, p. 75
OBU4_LSL	4842	Object Buffer 4 Linear Start Address Low	POSTED	4.3.10, p. 76
OBU4_LSH	4843	Object Buffer 4 Linear Start Address High	POSTED	4.3.10, p. 76
OBU4_BSX	4844	Object Buffer 4 X Size	POSTED	4.3.10.1, p. 77
OBU4_BSY	4845	Object Buffer 4 Y Size	POSTED	4.3.10.1, p. 77
OBU4_DEC	4846	Object Buffer 4 Decimate Control	POSTED	4.3.10.2, p. 78
OBU5_MCR	4850	Object Buffer 5 Master Control	POSTED	4.3.9.1, p. 74
OBU5_RFX	4851	Object Buffer 5 Reference Frame X Size	POSTED	4.3.9.2, p. 75
OBU5_LSL	4852	Object Buffer 5 Linear Start Address Low	POSTED	4.3.10, p. 76
OBU5_LSH	4853	Object Buffer 5 Linear Start Address High	POSTED	4.3.10, p. 76
OBU5_BSX	4854	Object Buffer 5 X Size	POSTED	4.3.10.1, p. 77
OBU5_BSY	4855	Object Buffer 5 Y Size	POSTED	4.3.10.1, p. 77
OBU5_DEC	4856	Object Buffer 5 Decimate Control	POSTED	4.3.10.2, p. 78
OBU6_MCR	4860	Object Buffer 6 Master Control	POSTED	4.3.9.1, p. 74
OBU6_RFX	4861	Object Buffer 6 Reference Frame X Size	POSTED	4.3.9.2, p. 75
OBU6_LSL	4862	Object Buffer 6 Linear Start Address Low	POSTED	4.3.10, p. 76
OBU6_LSH	4863	Object Buffer 6 Linear Start Address High	POSTED	4.3.10, p. 76
OBU6_BSX	4864	Object Buffer 6 X Size	POSTED	4.3.10.1, p. 77
OBU6_BSY	4865	Object Buffer 6 Y Size	POSTED	4.3.10.1, p. 77
OBU6_DEC	4866	Object Buffer 6 Decimate Control	POSTED	4.3.10.2, p. 78
OBU7_MCR	4870	Object Buffer 7 Master Control	POSTED	4.3.9.1, p. 74
OBU7_RFX	4871	Object Buffer 7 Reference Frame X Size	POSTED	4.3.9.2, p. 75
OBU7_LSL	4872	Object Buffer 7 Linear Start Address Low	POSTED	4.3.10, p. 76
OBU7_LSH	4873	Object Buffer 7 Linear Start Address High	POSTED	4.3.10, p. 76
OBU7_BSX	4874	Object Buffer 7 X Size	POSTED	4.3.10.1, p. 77



#### 4.3.7 RFU: Reference Frame Unit (cont.)

Name	Index	Definition	Posted?	Ref. Section
OBU7_BSY	4875	Object Buffer 7 Y Size	POSTED	4.3.10.1, p. 78
OBU7_DEC	4876	Object Buffer 7 Decimate Control	POSTED	4.3.10.2, p. 78
<b>DWU: Display Window Unit</b>			POSTED	4.3.11, p. 78
DWU_MCR	4100	Display Window Master Control	POSTED	4.3.11.1, p. 78
DWU_HCR	4101	Display Window Horizontal Control	POSTED	4.3.11.2, p. 80
DWU0_DZF	4400	Display Window 0 Zoom Factor	POSTED	4.3.11.3, p. 81
DWU0_RFX	4401	Display Window 0 Reference Frame X Size	POSTED	4.3.11.4, p. 81
DWU0_LSL	4402	Display Window 0 LSA Low	POSTED	4.3.11.5, p. 82
DWU0_LSH	4403	Display Window 0 LSA High	POSTED	4.3.11.5, p. 82
DWU0_WSX	4404	Display Window 0 X Size	POSTED	4.3.11.6, p. 82
DWU0_WSY	4405	Display Window 0 Y Size	POSTED	4.3.11.6, p. 82
DWU0_DSX	4406	Display Window 0 X Start	POSTED	4.3.11.7, p. 83
DWU0_DSY	4407	Display Window 0 Y Start	POSTED	4.3.11.7, p. 83
DWU1_DZF	4410	Display Window 1 Zoom Factor	POSTED	4.3.11.3, p. 81
DWU1_RFX	4411	Display Window 1 Reference Frame X Size	POSTED	4.3.11.4, p. 81
DWU1_LSL	4412	Display Window 1 LSA Low	POSTED	4.3.11.5, p. 82
DWU1_LSH	4413	Display Window 1 LSA High	POSTED	4.3.11.5, p. 82
DWU1_WSX	4414	Display Window 1 X Size	POSTED	4.3.11.6, p. 82
DWU1_WSY	4415	Display Window 1 Y Size	POSTED	4.3.11.6, p. 82
DWU1_DSX	4416	Display Window 1 X Start	POSTED	4.3.11.7, p. 83
DWU1_DSY	4417	Display Window 1 Y Start	POSTED	4.3.11.7, p. 83
DWU2_DZF	4420	Display Window 2 Zoom Factor	POSTED	4.3.11.3, p. 81
DWU2_RFX	4421	Display Window 2 Reference Frame X Size	POSTED	4.3.11.4, p. 81
DWU2_LSL	4422	Display Window 2 LSA Low	POSTED	4.3.11.5, p. 82
DWU2_LSH	4423	Display Window 2 LSA High	POSTED	4.3.11.5, p. 82
DWU2_WSX	4424	Display Window 2 X Size	POSTED	4.3.11.6, p. 82
DWU2_WSY	4425	Display Window 2 Y Size	POSTED	4.3.11.6, p. 82
DWU2_DSX	4426	Display Window 2 X Start	POSTED	4.3.11.7, p. 83



#### 4.3.7 RFU: Reference Frame Unit (cont.)

Name	Index	Definition	Posted?	Ref. Section
DWU2_DSY	4427	Display Window 2 Y Start	POSTED	4.3.11.7, p. 83
DWU3_DZF	4430	Display Window 3 Zoom Factor	POSTED	4.3.11.3, p. 81
DWU3_RFX	4431	Display Window 3 Reference Frame X Size	POSTED	4.3.11.4, p. 81
DWU3_LSL	4432	Display Window 3 LSA Low	POSTED	4.3.11.5, p. 82
DWU3_LSH	4433	Display Window 3 LSA High	POSTED	4.3.11.5, p. 82
DWU3_WSX	4434	Display Window 3 X Size	POSTED	4.3.11.6, p. 82
DWU3_WSY	4435	Display Window 3 Y Size	POSTED	4.3.11.6, p. 82
DWU3_DSX	4436	Display Window 3 X Start	POSTED	4.3.11.7, p. 83
DWU3_DSY	4437	Display Window 3 Y Start	POSTED	4.3.11.7, p. 83

#### 4.3.8 MMU: Memory Management Unit

##### 4.3.8.1 MMU\_MCR: MMU Master Control

I/O Address HIU\_RDT  
Index 4000

Register MMU\_MCR specifies the characteristics of the frame buffer used by the DVP.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD											FBD	FBC			

Bit #	Access	Reset	Description
15:5	R/W	0h	RSVD Reserved (read as '0').
4	R/W	0	FBD Frame Buffer Data Bus Width. 0 16-bit 1 32-bit
3:0	R/W	0000	FBC Frame Buffer Memory Device Address Configuration. 0000 64K 0001 128K 0010 256K 0011 1 M

ATI019099



### 4.3.9 OBU: Object Buffer Unit

#### 4.3.9.1 OBUo\_MCR: Object Buffer Master Control

POSTED

I/O Address	HIU_RDT
Index	4800 (OBU0_MCR: Object Buffer 0 Master Control)
	4810 (OBU1_MCR: Object Buffer 1 Master Control)
	4820 (OBU2_MCR: Object Buffer 2 Master Control)
	4830 (OBU3_MCR: Object Buffer 3 Master Control)
	4840 (OBU4_MCR: Object Buffer 4 Master Control)
	4850 (OBU5_MCR: Object Buffer 5 Master Control)
	4860 (OBU6_MCR: Object Buffer 6 Master Control)
	4870 (OBU7_MCR: Object Buffer 7 Master Control)

The eight identical registers OBUo\_MCR control the operation of the eight object buffers.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD			LME	CME	FL	OPM				SSM	YBDC	XBDC	FA		

Bit #	Access	Reset	Description
15:13	R/W	000	RSVD    Reserved (read as '0').
12	R/W	0	LME    Luminance Mask Enable. Specifies whether the MSB of a 16-bit input stream (typically the Y channel of YCbCr data) is written to the object buffer or masked. 0      Enable luminance data update (MSB written to object buffer) 1      Disable luminance data update (MSB masked)
11	R/W	0	CME    Chrominance Mask Enable. Specifies whether the LSB of a 16-bit input stream (typically the CbCr channel of YCbCr data) is written to the object buffer or masked. 0      Enable chrominance data update (LSB written to object buffer) 1      Disable chrominance data update (LSB masked)
10	R/W	0	FL      Field Lock. Field-locks the object to the video source selected as the master in register VIU_WDT, bit MFTS. 0      not field locked 1      field locked
9:6	R/W	0h	OPM    Operation Mode. Enables operation of the object buffer and specifies the synchronization and addressing modes. 0000   Disable OBU line 0001   Enable OBU; lock to IPU1, address generation locked to IPU1 (bit FL must = 1) 0100   Enable OBU; independent, interlaced addresses, start on line 1 0101   Enable OBU; independent, interlaced addresses, start on line 2 1100   Enable OBU; independent, normal addresses 1101   Enable OBU; independent, line replicate addresses (on read) 1110   Enable OBU; independent, block mode addresses (8 x 8 blocks, OBU0 only) 1111   Enable 16 x 8 blocks. OBU0 only XXXX   All other configurations reserved.

ATI019100



Bit #	Access	Reset	Description
5	R/W	0	SSM Single Sweep Mode. 0 Disable single sweep mode 1 Enable single sweep mode (reset OPM to 00000 after one field)
4	R/W	0	YBDC Y BLT Direction Control. Specifies whether the Y address counter is incremented or decremented after each line. 0 BLT to decreasing memory addresses 1 BLT to increasing memory addresses
3	R/W	0	XBDC X BLT Direction Control. Specifies whether the X address counter is incremented or decremented after each line. 0 BLT to decreasing memory addresses 1 BLT to increasing memory addresses
2:0	R/W	000	FA FIFO Association. Specifies whether the stream written into the object buffer is to be copied to one of the output FIFOs. 000 No FIFO copy 001 Copy object buffer to FIFO A during write 010 Copy object buffer to FIFO B during write 011 Copy object buffer to FIFO C during write 100 Copy object buffer to FIFO D during write XXX All other configurations reserved

#### 4.3.9.2 OBUo\_RFX: Object Buffer Reference Frame Size

POSTED

I/O Address	HIU_RDT
Index	4801 (OBU0_RFX: Object Buffer 0 Reference Frame X Size) 4811 (OBU1_RFX: Object Buffer 1 Reference Frame X Size) 4821 (OBU2_RFX: Object Buffer 2 Reference Frame X Size) 4831 (OBU3_RFX: Object Buffer 3 Reference Frame X Size) 4841 (OBU4_RFX: Object Buffer 4 Reference Frame X Size) 4851 (OBU5_RFX: Object Buffer 5 Reference Frame X Size) 4861 (OBU6_RFX: Object Buffer 6 Reference Frame X Size) 4871 (OBU7_RFX: Object Buffer 7 Reference Frame X Size)

The eight identical registers OBUo\_RFX specify, for each of the eight object buffers, the 11-bit width (in pixels) of the reference frame containing the object buffer.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						RFX									

Bit #	Access	Reset	Description
15:11	R/W	0h	RSVD Reserved (read as '0').
10:0	R/W	0h	RFX Reference Frame X size (0-7FFh)

ATI019101



#### 4.3.10 OBUo\_LSB: Object Buffer Linear Start Address

POSTED

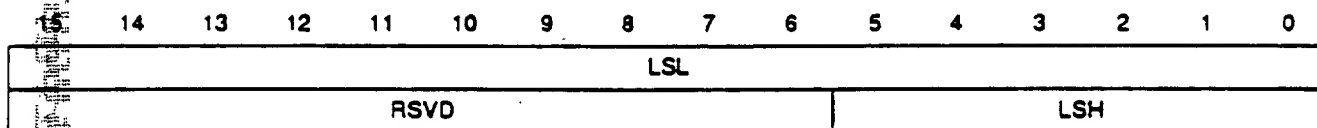
I/O Address

HIU\_RDT

Index

4802 (OBU0\_LSL: Object Buffer 0 Linear Start Address Low)  
4812 (OBU1\_LSL: Object Buffer 1 Linear Start Address Low)  
4822 (OBU2\_LSL: Object Buffer 2 Linear Start Address Low)  
4832 (OBU3\_LSL: Object Buffer 3 Linear Start Address Low)  
4842 (OBU4\_LSL: Object Buffer 4 Linear Start Address Low)  
4852 (OBU5\_LSL: Object Buffer 5 Linear Start Address Low)  
4862 (OBU6\_LSL: Object Buffer 6 Linear Start Address Low)  
4872 (OBU7\_LSL: Object Buffer 7 Linear Start Address Low)  
4803 (OBU0\_LSH: Object Buffer 0 Linear Start Address High)  
4813 (OBU1\_LSH: Object Buffer 1 Linear Start Address High)  
4823 (OBU2\_LSH: Object Buffer 2 Linear Start Address High)  
4833 (OBU3\_LSH: Object Buffer 3 Linear Start Address High)  
4843 (OBU4\_LSH: Object Buffer 4 Linear Start Address High)  
4853 (OBU5\_LSH: Object Buffer 5 Linear Start Address High)  
4863 (OBU6\_LSH: Object Buffer 6 Linear Start Address High)  
4873 (OBU7\_LSH: Object Buffer 7 Linear Start Address High)

Registers OBUo\_LSL and OBUo\_LSH specify the 23-bit linear starting address of the object buffer.



Bit	Access	Reset	Description
-----	--------	-------	-------------

##### Object Buffer Linear Start Address Low

15:0	R/W	0h	LSL	Linear Start Address Low. Specifies the lower bits of the 22-bit linear starting address (LSb must = 0). (0-FFFEh)
------	-----	----	-----	--

##### Object Buffer Linear Start Address High

15:6	R/W	0h	RSVD	Reserved (read as '0').
5:0	R/W	0h	LSH	Linear Start Address High. Specifies the upper 6 bits of the 22-bit linear starting address. (0-7Fh)

ATI019102



#### 4.3.10.1 OBUo\_BSa: Object Buffer Size

POSTED-

I/O Address HIU\_RDT

Index	4804 (OBU0_BSY: Object Buffer 0 X Size)	4805 (OBU0_BSY: Object Buffer 0 Y Size)
	4814 (OBU1_BSY: Object Buffer 1 X Size)	4815 (OBU1_BSY: Object Buffer 1 Y Size)
	4824 (OBU2_BSY: Object Buffer 2 X Size)	4825 (OBU2_BSY: Object Buffer 2 Y Size)
	4834 (OBU3_BSY: Object Buffer 3 X Size)	4835 (OBU3_BSY: Object Buffer 3 Y Size)
	4844 (OBU4_BSY: Object Buffer 4 X Size)	4845 (OBU4_BSY: Object Buffer 4 Y Size)
	4854 (OBU5_BSY: Object Buffer 5 X Size)	4855 (OBU5_BSY: Object Buffer 5 Y Size)
	4864 (OBU6_BSY: Object Buffer 6 X Size)	4865 (OBU6_BSY: Object Buffer 6 Y Size)
	4874 (OBU7_BSY: Object Buffer 7 X Size)	4875 (OBU7_BSY: Object Buffer 7 Y Size)

Registers OBUo\_BSY and OBUo\_BSY specify the size of the object buffer.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						BSX									
RSVD						BSY									

Bit # Access Reset Description

#### OBUo\_BSY: Object Buffer X Size

15:11	R/W	0h	RSVD	Reserved (read as '0').
10:0	R/W	0h	BSX	Buffer X Size. Specifies the object buffer's width in pixels. The hardware always forces the LSb to '0'. (0-7FFh)

#### OBUo\_BSY: Object Buffer Y Size

15:11	R/W	0h	RSVD	Reserved (read as '0').
10:0	R/W	0h	BSY	Buffer Y Size. Specifies the object buffer's height in pixels. (0-7FFh)

ATI019103





#### 4.3.10.2 OBUo\_DEC: Object Buffer Decimate Control

POSTED

I/O Address HIU\_RDT

Index 4806 (OBU0\_DEC: Object Buffer 0 Decimate Control)  
4816 (OBU1\_DEC: Object Buffer 1 Decimate Control)  
4826 (OBU2\_DEC: Object Buffer 2 Decimate Control)  
4836 (OBU3\_DEC: Object Buffer 3 Decimate Control)  
4846 (OBU4\_DEC: Object Buffer 4 Decimate Control)  
4856 (OBU5\_DEC: Object Buffer 5 Decimate Control)  
4866 (OBU6\_DEC: Object Buffer 6 Decimate Control)  
4876 (OBU7\_DEC: Object Buffer 7 Decimate Control)

Register OBUo\_DEC specifies the write decimation mask. DM7-DM0 are mapped to each successive group of eight pixels written into the object buffer. Do not drop the first line or pixel in a transfer to an object buffer (i.e., when BLT direction is up, DM0 must be '0'; when BLT direction is down, DM7 must be '0').

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DM7	DM6	DM5	DM4	DM3	DM2	DM1	DM0

Bit #	Access	Reset	Description
15:8	R/W	0h	RSVD Reserved (read as '0').
7	R/W	0h	DM7 Write Decimation Mask Bit 7.
6	R/W	0h	DM6 Write Decimation Mask Bit 6.
5	R/W	0h	DM5 Write Decimation Mask Bit 5.
4	R/W	0h	DM4 Write Decimation Mask Bit 4.
3	R/W	0h	DM3 Write Decimation Mask Bit 3.
2	R/W	0h	DM2 Write Decimation Mask Bit 2.
1	R/W	0h	DM1 Write Decimation Mask Bit 1.
0	R/W	0h	DM0 Write Decimation Mask Bit 0.

For all Write Decimation Bits 7:0:  
0 Write pixel to frame buffer  
1 Drop pixel

#### 4.3.11 DWU: Display Window Unit

##### 4.3.11.1 DWU\_MCR: Display Window Master Control

POSTED

I/O Address HIU\_RDT

Index 4100

Register DWU\_MCR controls the operation of the display window and indicates to the RFU whether or not the CL-PX2080 is present.

ATI019104

**CL-PX2070**  
Digital Video Processor



**Pixel Semiconductor**  
A Cirrus Logic Company

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GCS	GFP	GFM	GVSP	GHSP	GBP	OCC	IMS	RSVD				WC3	WC2	WC1	WC0

**Bit # Access Reset Description**

15	R/W	0	GCS	Graphics Clock Select 0 1/2x GPCLK 1 1x GPCLK
14	R/W	0	GFP	Graphics Field Polarity 0 normal polarity 1 inverted polarity
13	R/W	0	GFM	Graphics Field Mode 0 field polarity determined by value of GHSP on falling GVSP 1 GHSP input used as field select
12	R/W	0	GVSP	Graphics Vertical Sync Polarity. Specifies polarity of signal GVS. 0 active low 1 active high
11	R/W	0	GHSP	Graphics Horizontal Sync Polarity. Specifies polarity of signal GHS. 0 active low 1 active high
10	R/W	0	GBP	Graphics Blank Polarity. Specifies polarity of signal GBL. 0 active low 1 active high
9	R/W	0	OCC	Occluded Window Control. Specifies whether the present hardware configuration includes the CL-PX2080. 0 CL-PX2080 is present — system supports occluded windows 1 CL-PX2080 is not present — system does not support occluded windows
8	R/W	0	IMS	Interlace Mode Select. Specifies whether the stream stored in the object buffer for display by the current display window is interlaced or non-interlaced. 0 Progressive-scan video (non-interlaced) 1 Interlaced video
7:4	R/W	0000	RSVD	Reserved (read as '0').
3	R/W	0	WC3	Window 3 Control
2	R/W	0	WC2	Window 2 Control
1	R/W	0	WC1	Window 1 Control
0	R/W	0	WC0	Window 0 Control

For all Window Controls 3:0:  
0 Disable window  
1 Enable window

ATI019105



#### 4.3.11.2 DWU\_HCR: Display Window Horizontal Control

POSTED

I/O Address HIU\_RDT

Index 4101

Register DWU\_HCR shares two functions, depending on whether or not the DVP is operating with the CL-PX2080, as specified register DWU\_MCR, bit OCC.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD					HAC										
RSVD										MWS					

Bit #	Access	Reset	Description
-------	--------	-------	-------------

Horizontal Active Count (DWU\_MCR, bit OCC = 0)

15:11	R/W	0h	RSVD	Reserved (read as '0').
10:0	R/W	0h	HAC	Horizontal Active Count. Specifies the number of pixel periods in the horizontal line active interval for the output CRT display. (0-7FFh)

Minimum Window Separation (DWU\_MCR, bit OCC = 1)

15:8	R/W	0h	RSVD	Reserved (read as '0').
7:0	R/W	0h	MWS	Minimum Window Separation. Specifies the minimum number of pixel periods required to separate display windows. (0-ffh)

ATI019106



#### 4.3.11.3 DWUd\_DZF: Display Window Display Zoom Factor

POSTED

I/O Address	HIU_RDT
Index	4400 (DWU0_DZF: Display Window 0 Zoom Factor)
	4410 (DWU1_DZF: Display Window 1 Zoom Factor)
	4420 (DWU2_DZF: Display Window 2 Zoom Factor)
	4430 (DWU3_DZF: Display Window 3 Zoom Factor)

Register DWUd\_DZF specifies the X and Y zoom factors to be applied to the display window output (functional only when used with CL-PX2080). The image is scaled according to the following formula:

$$\text{Scaling} = \frac{256}{\text{ZOOM FACTOR}}$$

For example, a zoom factor of 128 yields a scaling factor of 2. A zoom factor of '0h' specifies a scaling factor of one (no change in image size).

**NOTE:** The contents of the object buffer are not affected by the zoom factors.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
YZOOM								XZOOM							

Bit #	Access	Reset	Description
15:8	R/W	0h	YZOOM Y Zoom Factor — line replication value. (0-FFh)
7:0	R/W	0h	XZOOM X Zoom Factor — pixel replication value. (0-FFh)

#### 4.3.11.4 DWUd\_RFX: Display Window Reference Frame Size

POSTED

I/O Address	HIU_RDT
Index	4401 (DWU0_RFX: Display Window 0 Reference Frame X Size)
	4411 (DWU1_RFX: Display Window 1 Reference Frame X Size)
	4421 (DWU2_RFX: Display Window 2 Reference Frame X Size)
	4431 (DWU3_RFX: Display Window 3 Reference Frame X Size)

Register DWUd\_RFX specifies the 11-bit pixel width of the reference frame containing the display window.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						RFX									

Bit #	Access	Reset	Description
15:11	R/W	0h	RSVD Reserved (read as '0').
10:0	R/W	0h	RFX Reference Frame X size. (0-7FFh)

ATI019107 =



#### 4.3.11.5 DWUd\_LSB: Display Window Linear Start Address

POSTED

I/O Address HIU\_RDT

Index 4402 (DWU0\_LSL: Display Window 0 LSA Low) 4403 (DWU0\_LSH: Display Window 0 LSA High)  
4412 (DWU1\_LSL: Display Window 1 LSA Low) 4413 (DWU1\_LSH: Display Window 1 LSA High)  
4422 (DWU2\_LSL: Display Window 2 LSA Low) 4423 (DWU2\_LSH: Display Window 2 LSA High)  
4432 (DWU3\_LSL: Display Window 3 LSA Low) 4433 (DWU3\_LSH: Display Window 3 LSA High)

Registers DWUd\_LSL and DWUd\_LSH specify the 23-bit linear starting address of the display window.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LSL															
RSVD										LSH					

Bit #	Access	Reset	Description
-------	--------	-------	-------------

#### DWUd\_LSL: Display Window Linear Start Address Low

15:0	R/W	0h	LSL	Linear Start Address Low. Specifies the lower bits of the 22-bit linear starting address. (LSb must = 0). (0-7FFFh)
------	-----	----	-----	---

#### DWUd\_LSH: Display Window Linear Start Address High

15:7	R/W	0h	RSVD	Reserved (read as '0').
6:0	R/W	0h	LSH	Linear Start Address High. Specifies the upper 7 bits of the 22-bit linear starting address. (0-7Fh)

#### 4.3.11.6 DWUd\_WSa: Display Window Size

POSTED

I/O Address HIU\_RDT

Index 4404 (DWU0\_WSX: Display Window 0 X Size) 4405 (DWU0\_WSY: Display Window 0 Y Size)  
4414 (DWU1\_WSX: Display Window 1 X Size) 4415 (DWU1\_WSY: Display Window 1 Y Size)  
4424 (DWU2\_WSX: Display Window 2 X Size) 4426 (DWU2\_WSY: Display Window 2 Y Size)  
4434 (DWU3\_WSX: Display Window 3 X Size) 4435 (DWU3\_WSY: Display Window 3 Y Size)

Registers DWUd\_WSX and DWUd\_WSY specify the size of the display window.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD						WSX									
RSVD						WSY									

ATI019108



Bit #	Access	Reset	Description
-------	--------	-------	-------------

DWUd\_WSX: Display Window X Size

15:11	R/W	0h	RSVD	Reserved (read as '0').
10:0	R/W	0h	WSX	Window X Size. Specifies the X dimension of the display window in pixels. (LSb must = 0)

DWUd\_WSY: Display Window Y Size

15:11	R/W	0h	RSVD	Reserved (read as '0').
10:0	R/W	0h	WSY	Window Y Size. Specifies the Y dimension of the display window in pixels. (0-7FFh)

#### 4.3.11.7 DWUd\_DSa: Display Window Start

POSTED

I/O Address	HIU_RDT
Index	
4406 (DWU0_DSX: Display Window 0 X Start)	4407 (DWU0_DSY: Display Window 0 Y Start)
4416 (DWU1_DSX: Display Window 1 X Start)	4417 (DWU1_DSY: Display Window 1 Y Start)
4426 (DWU2_DSX: Display Window 2 X Start)	4427 (DWU2_DSY: Display Window 2 Y Start)
4436 (DWU3_DSX: Display Window 3 X Start)	4437 (DWU3_DSY: Display Window 3 Y Start)

Registers DWUd\_DSX and DWUd\_DSY specify the location of the top left corner of the display window relative to the top left corner of the output CRT display.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD								DSX							
RSVD								DSY							

Bit #	Access	Reset	Description
-------	--------	-------	-------------

DWUd\_DSX: Display Window X Start

15:12	R/W	0h	RSVD	Reserved (read as '0').
11:0	R/W	0h	DSX	Display X Start. Specifies the pixel offset from the CRT column 0 to the left-most column of the display window. (0-7FFh)

DWUd\_DSY: Display Window Y Start

15:12	R/W	0h	RSVD	Reserved (read as '0').
11:0	R/W	0h	DSY	Display Y Start. Specifies the pixel offset from the CRT row 0 to the top-most row of the display window. (0-7FFh)

ATI019109



## 5. ELECTRICAL SPECIFICATIONS

### 5.1 Absolute Maximum Ratings

This section lists the absolute maximum ratings of the DVP. Stresses above those listed can cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these or any conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods can affect device reliability.

Storage temperature..... -65 to +150°C  
Voltage on any pin with respect to ground..... -0.5 Volts to  $V_{DD} + 0.5V$   
Power Supply Voltage ..... 7V  
Lead Temperature (10 seconds)..... 300°C

### 5.2 DVP Specifications (Digital)

Symbol	Parameter	MIN	MAX	Conditions
$V_{DD}$	Power Supply Voltage	4.75 V	5.25 V	Normal Operation
$V_{IL}$	Input Low Voltage	0 V	0.8 V	
$V_{IH}$	Input High Voltage	2.0 V	$V_{DD} + 0.8 V$	
$V_{OL}$	Output Low Voltage		0.4 V	$I_{OL} = 4 \text{ mA}$
$V_{OH}$	Output High Voltage	2.4 V	V	$I_{OH} = 400 \mu\text{A}$
$I_{DD}$	Digital Supply Current		N/A	$V_{DD}$ Nominal
$I_{LIS}$	Input Leakage	-10 $\mu\text{A}$	10 $\mu\text{A}$	$0 < V_{IN} < V_{DD}$
$C_{IN}$	Input Capacitance		10 pF	
$C_{OUT}$	Output Capacitance		10 pF	



### 5.3 AC Characteristics/Timing Information

This section includes system timing requirements for the DVP. Timings are provided in nanoseconds (ns), at TTL input levels, with the ambient temperature varying from 0 to 70°C, and  $V_{CC}$  varying from 4.75 to 5.25V DC.

- NOTE:**
1. All timings assume a load of 50 pF.
  2. TTL signals are measured at TTL threshold; CMOS signals are measured at CMOS threshold.

#### 5.3.1 Index of Timing Information

<b>ISA Bus Timing</b> .....	<b>86</b>
Figure 5-1.    ISA Bus — I/O Timing .....	87
<b>MCA Bus Timing</b> .....	<b>88</b>
Figure 5-2.    MCA Bus — I/O Timing .....	89
Figure 5-3.    MCA Bus — CDSFDBK* Timing .....	89
Figure 5-4.    MCA Bus — CDSETUP* Timing .....	89
<b>Local Hardware Interface Timing</b> .....	<b>90</b>
Figure 5-5.    Local Hardware Interface — Write Timing .....	90
Figure 5-6.    Local Hardware Interface — Read Timing .....	91
<b>Video Port Timing</b> .....	<b>92</b>
Figure 5-7.    Video I/O Timing .....	93
Figure 5-8.    STALL* and STALLRQ* Timing .....	93
Figure 5-9.    Video and Graphics Port Timing .....	93
<b>Memory Timing</b> .....	<b>94</b>
Figure 5-10.   Read Transfer Cycle Timing .....	94
Figure 5-11.   CAS* Before RAS* Refresh Timing .....	95
Figure 5-12.   Memory Read Timing .....	97
Figure 5-13.   Memory Write Timing .....	97

ATI019111



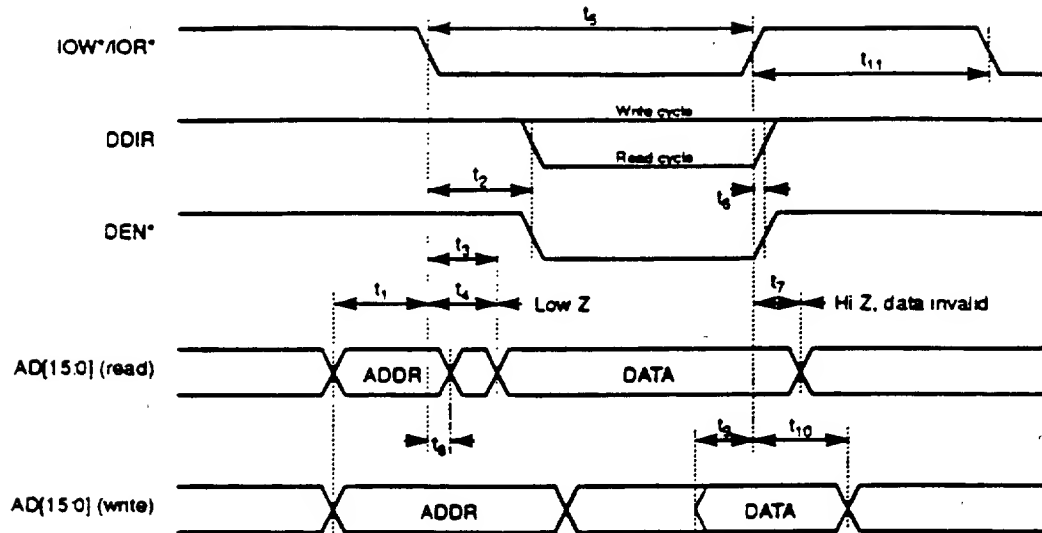


### 5.3.2 ISA Bus Timing

Table 5-1. ISA Bus Timing

Ref.	Parameter	MIN	MAX
t <sub>1</sub>	Setup AD[15:0] address valid before IOR*/IOW* active	30 ns	
t <sub>2</sub>	Delay IOR*/IOW* active to DEN* active, DDIR change	4 ns	20 ns
t <sub>3</sub>	Delay IOR* active to AD[15:0] read data out low Z	4 ns	75 ns
t <sub>4</sub>	Delay IOR* active to AD[15:0] read data out valid		75 ns
t <sub>5</sub>	Pulse Width IOR*/IOW*	100 ns	
t <sub>6</sub>	Delay IOR*/IOW* inactive to DEN* inactive, DDIR change	4 ns	20 ns
t <sub>7</sub>	Delay IOR* inactive to AD[15:0] read data invalid	4 ns	20 ns
t <sub>8</sub>	Hold AD[15:0] address valid after IOR*/IOW* active	4 ns	
t <sub>9</sub>	Setup AD[15:0] write data valid before IOW* inactive	50 ns	
t <sub>10</sub>	Hold AD[15:0] write data valid after IOW* inactive	4 ns	
t <sub>11</sub>	Delay IOW*/IOR* inactive to IOW*/IOR* active	80 ns	

ATI019112



NOTE: AEN must be low during cycle.

Figure 5-1. ISA Bus — I/O Timing

ATI019113



### 5.3.3 MCA Bus Timing

Table 5-2. MCA Bus Timing

Ref.	Parameter	MIN	MAX
t <sub>1</sub>	Setup AD[15:0] address valid before ADL* active	40 ns	
t <sub>2</sub>	Setup S0*, S1* valid before ADL* active	7 ns	
t <sub>3</sub>	Pulse Width ADL*	35 ns	
t <sub>4</sub>	Hold S0*, S1* from ADL* inactive	20 ns	
t <sub>5</sub>	Hold AD[15:0] address from ADL* inactive	25 ns	
t <sub>6</sub>	Hold MIO* from ADL* inactive		
t <sub>7</sub>	Setup AD[15:0] address valid before CMD* active	80 ns	
t <sub>8</sub>	Setup S0*, S1* valid before CMD* active	50 ns	
t <sub>9</sub>	Setup ADL* active before CMD* active	35 ns	
t <sub>10</sub>	Hold AD[15:0] address from CMD* active	25 ns	
t <sub>11</sub>	Hold S0*, S1* from CMD* active	25 ns	
t <sub>12</sub>	Setup AD[15:0] write data valid before CMD* active	15 ns	
t <sub>13</sub>	Hold AD[15:0] write data valid from CMD* inactive	0 ns	
t <sub>14</sub>	Delay CMD* active to AD[15:0] read data valid	45 ns	
t <sub>15</sub>	Delay CMD* inactive to AD[15:0] read data invalid	0 ns	
t <sub>16</sub>	Delay CMD* inactive to AD[15:0] read data high Z		30 ns
t <sub>17</sub>	Delay CMD* active to DEN* active/DDIR change		35 ns
t <sub>18</sub>	Delay CMD* inactive to DEN* inactive/DDIR change		20 ns
t <sub>19</sub>	Delay CMD* inactive to CMD* active		
t <sub>20</sub>	Pulse Width CMD*	90 ns	
t <sub>21</sub>	Delay AD[15:0] address, MIO* valid to CDSFDBK* active		55 ns
t <sub>22</sub>	Delay AD[15:0] address, MIO* invalid to CDSFDBK* inactive	0 ns	
t <sub>23</sub>	Setup CDSETUP* active before ADL* active	10 ns	
t <sub>24</sub>	Hold CDSETUP* active after CMD* active	25 ns	
t <sub>25</sub>	Hold CDSETUP* active after ADL* inactive	20 ns	

ATI019114

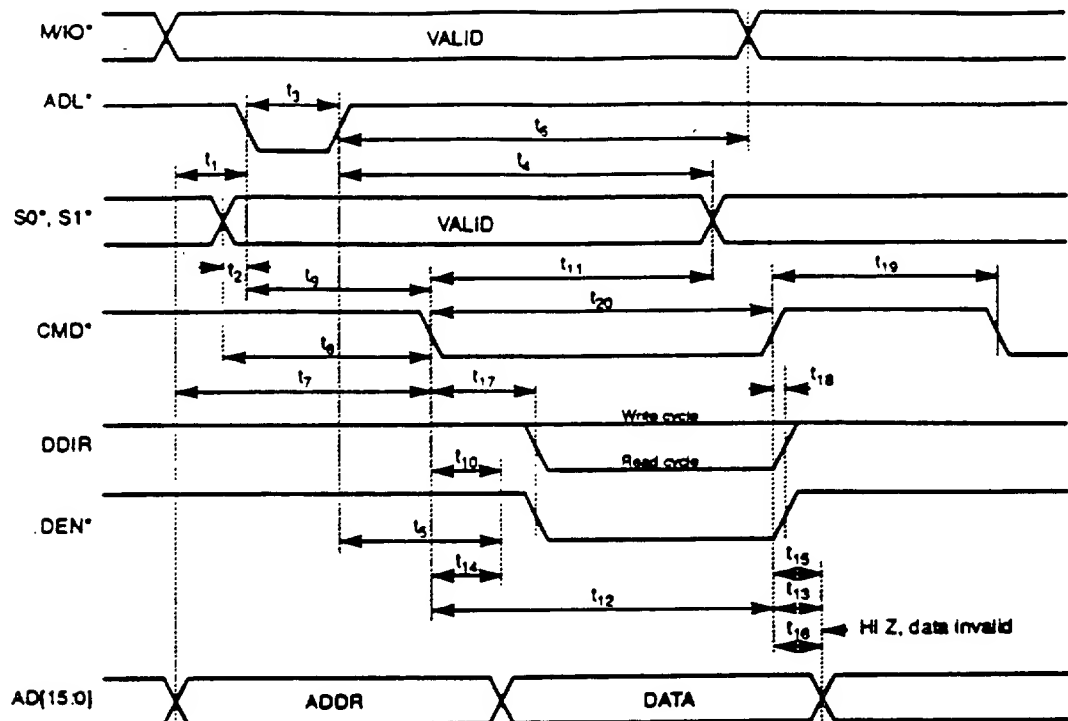
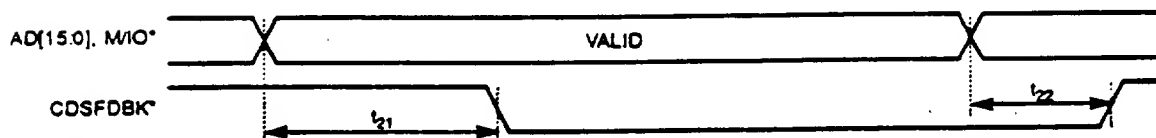


Figure 5-2. MCA Bus — I/O Timing



NOTE: Slaves do not drive CDSFDBK\* when they are selected by the 'card setup' signal.

Figure 5-3. MCA Bus — CDSFDBK\* Timing

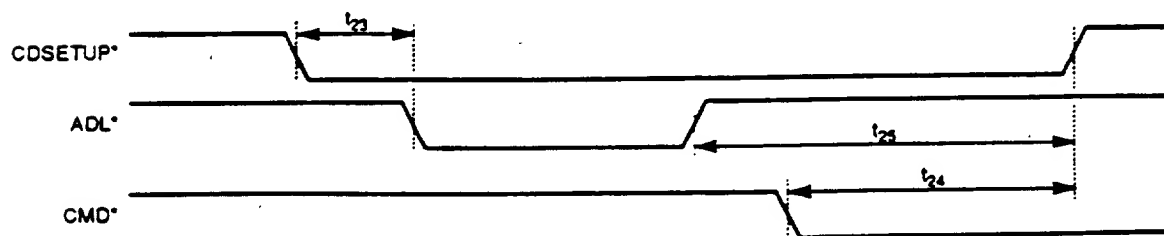


Figure 5-4. MCA Bus — CDSETUP\* Timing

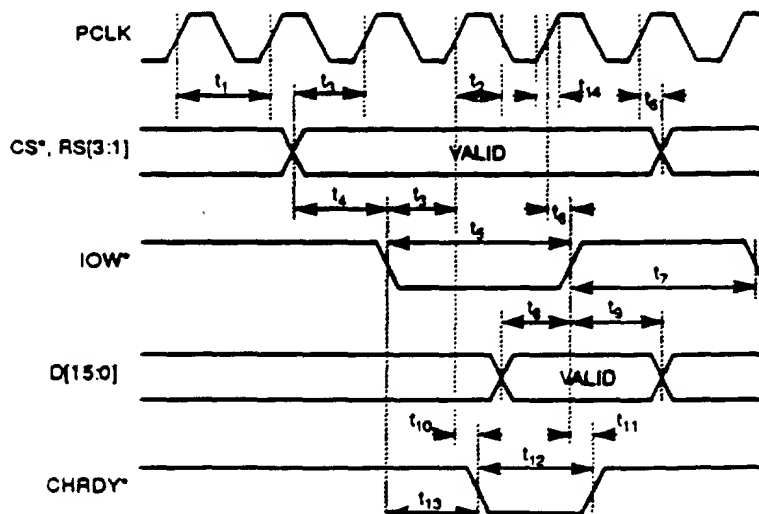
ATI019115



### 5.3.4 Local Hardware Interface Timing

Table 5-3. Local Hardware Interface — Write Timing

Ref.	Parameter	MIN	MAX
t <sub>1</sub>	Period PCLK	50 ns	
t <sub>2</sub>	Pulse Width PCLK	12 ns	
t <sub>3</sub>	Setup IOW*, CS* before PCLK rising edge	10 ns	
t <sub>4</sub>	Setup CS*, RS[3:1] before IOW* active	1 cycle	
t <sub>5</sub>	Pulse Width IOW*	2 cycles	
t <sub>6</sub>	Hold PCLK rising edge to IOW*, CS* transition	2 ns	
t <sub>7</sub>	Delay IOW* inactive to IOW* active	2 cycles	
t <sub>8</sub>	Setup D[15:0] valid before IOW* inactive	15 ns	
t <sub>9</sub>	Hold CS*, RS[3:1], D[15:0] valid to IOW* inactive	2 ns	
t <sub>10</sub>	Delay PCLK rising edge to CHRDY* active	4 ns	20 ns
t <sub>11</sub>	Delay IOW* inactive to CHRDY* inactive	4 ns	20 ns
t <sub>12</sub>	Pulse Width CHRDY*	1 cycle	2 cycles
t <sub>13</sub>	Delay IOW* active to CHRDY* active	1 cycle	1 cycle
t <sub>14</sub>	Transition PCLK		5 ns



**NOTES:** Timing is shown relative to clock. Internally, D[15:0], IOW\*/IOR\*, RS[3:1] must be stable for the entire cycle following CS\* active. D[15:0] is sampled on the 2nd rising edge after CS\* is asserted.

CS\*, IOW\*, RS[3:1] must be asserted.

If IOW\* exceeds 2 cycles, CHRDY\* is negated after 2 cycles. In this case, t<sub>11</sub> is referenced to PCLK.

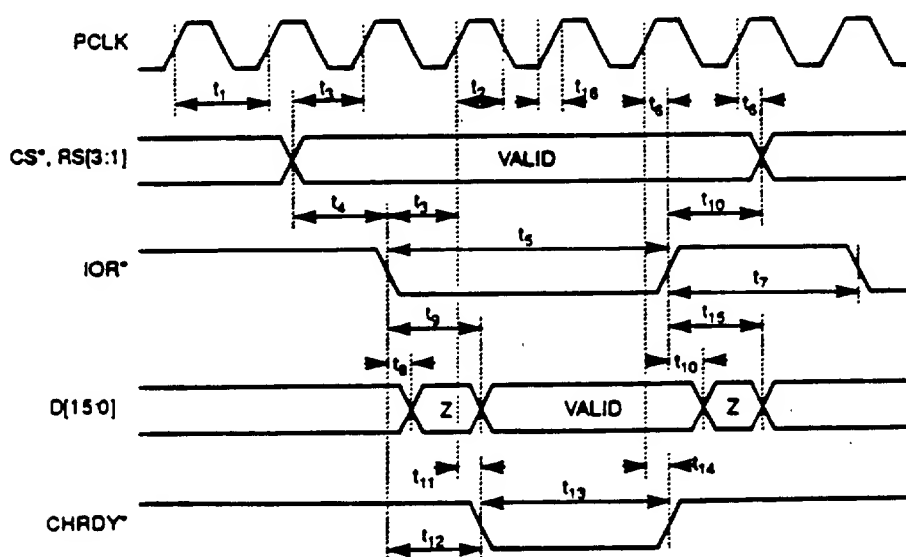
Figure 5-5. Local Hardware Interface — Write Timing

ATI019116



Table 5-4. Local Hardware Interface — Read Timing

Ref.	Parameter	MIN	MAX
t <sub>1</sub>	Period PCLK	50 ns	
t <sub>2</sub>	Pulse Width PCLK	12 ns	
t <sub>3</sub>	Setup IOR*, CS* active before PCLK rising edge	12 ns	
t <sub>4</sub>	Setup CS*, RS[3:1] valid before IOR* active	1 cycle	
t <sub>5</sub>	Pulse Width IOR*	3 cycles	
t <sub>6</sub>	Hold PCLK rising edge to IOR* inactive, CS* inactive	2 ns	
t <sub>7</sub>	Delay IOR* inactive to IOR*/IOW* active	2 cycles	
t <sub>8</sub>	Delay IOR* active to D[15:0] low impedance	4 ns	20 ns
t <sub>9</sub>	Delay IOR* active to D[15:0] valid	4 ns	40 ns
t <sub>10</sub>	Hold IOR* inactive to D[15:0], CS*, RS[3:1] invalid	2 ns	
t <sub>11</sub>	Delay PCLK rising edge to CHRDY* active	4 ns	20 ns
t <sub>12</sub>	Delay IOR* active to CHRDY* active	1 cycle	1 cycle
t <sub>13</sub>	Pulse Width CHRDY*	2 cycles	2 cycles
t <sub>14</sub>	Delay PCLK rising edge to CHRDY* inactive	4 ns	20 ns
t <sub>15</sub>	Delay IOR* inactive to D[15:0] high impedance	2 ns	20 ns
t <sub>16</sub>	Transition PCLK		5 ns



ATI019117

Figure 5-6. Local Hardware Interface — Read Timing



### 5.3.5 Video Port Timing

Table 5-5. Video Port Timing

Ref.	Parameter	MIN	MAX
t <sub>1</sub>	Period VnCLK	33 ns	
t <sub>2</sub>	Pulse width VnCLK high	12 ns	
t <sub>3</sub>	Setup VnPH before VnCLK rising edge	10 ns	
t <sub>4</sub>	Hold VnPH from VnCLK rising edge	2 ns	
t <sub>5</sub>	Delay VnD[15:0] output, VnVS/VnHS/VnBL valid after VnCLK rising edge	5 ns	15 ns
t <sub>6</sub>	Delay VnIEN* valid after VnCLK rising edge	5 ns	15 ns
t <sub>7</sub>	Transition GPCLK		5 ns
t <sub>8</sub>	Transition SBCLK		5 ns
t <sub>9</sub>	Setup VnD[15:0] input, VnVS/VnHS/VnBL before VnCLK rising edge	10 ns	
t <sub>10</sub>	Hold VnD[15:0] input, VnVS/VnHS/VnBL after VnCLK rising edge	2 ns	
t <sub>11</sub>	Transition VnCLK		5 ns
t <sub>12</sub>	Setup STALLRQ* active before V2CLK rising edge	10 ns	
t <sub>13</sub>	Hold STALLRQ* active after V2CLK rising edge	2 ns	
t <sub>14</sub>	Hold STALL* valid after V2CLK rising edge	7 ns	20 ns
t <sub>15</sub>	Hold STALL* invalid after V2CLK rising edge	7 ns	20 ns
t <sub>16</sub>	Pulse Width GPCLK high	4 ns	
t <sub>17</sub>	Pulse Width GPCLK low	4 ns	
t <sub>18</sub>	Period GPCLK	12.5 ns	12.5 ns
t <sub>19</sub>	Setup GHS, GVS, GBL before GPCLK rising edge	10 ns	
t <sub>20</sub>	Hold GHS, GVS, GBL after GPCLK rising edge	2 ns	
t <sub>21</sub>	Delay FCLK rising edge after GPCLK rising edge	5 ns	15 ns
t <sub>22</sub>	Delay SBCLK rising edge after GPCLK rising edge	5 ns	15 ns
t <sub>23</sub>	Delay SBCLK low from GPCLK rising edge	5 ns	15 ns
t <sub>24</sub>	Delay FCLK low from GPCLK rising edge	5 ns	15 ns
t <sub>25</sub>	Delay SBCLK rising edge from FCLK	0 ns	5 ns
t <sub>26</sub>	Setup FRDY valid before GPCLK rising edge	10 ns	
t <sub>27</sub>	Hold FRDY valid after GPCLK rising edge	2 ns	
t <sub>28</sub>	Setup ZC[3:0] valid before FCLK rising edge	10 ns	
t <sub>29</sub>	Hold ZC[3:0] valid after FCLK rising edge	2 ns	

ATI019118

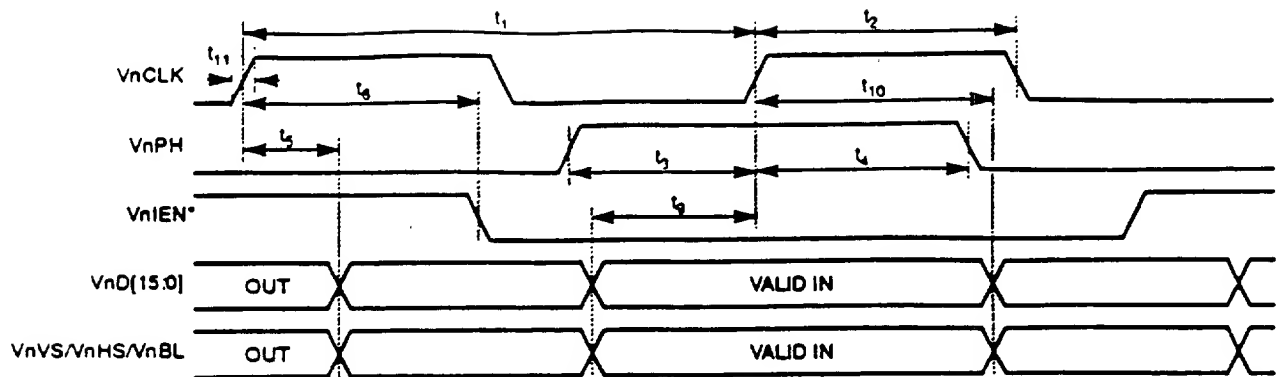


Figure 5-7. Video I/O Timing

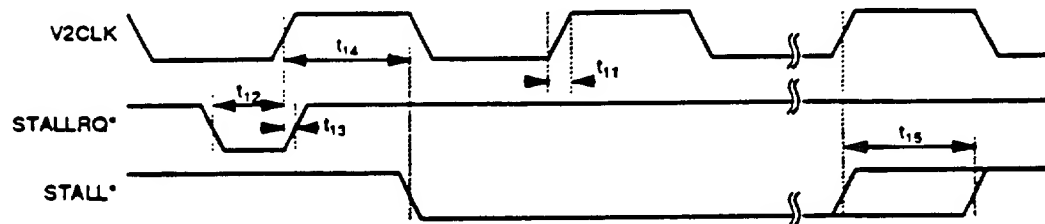


Figure 5-8. STALL\* and STALLRQ\* Timing

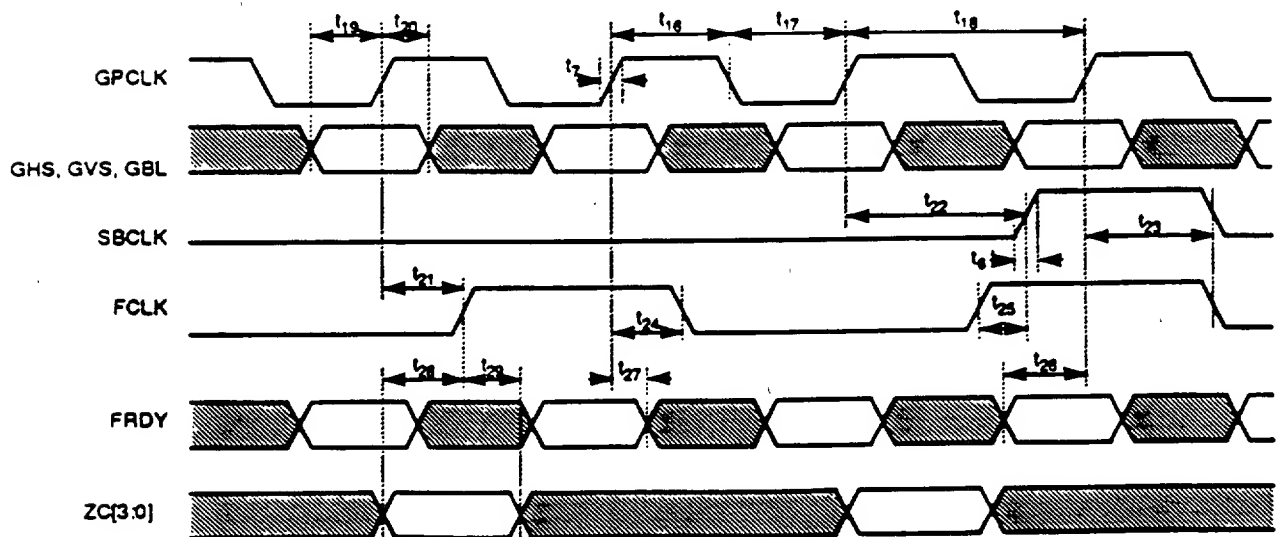


Figure 5-9. Video and Graphics Port Timing

ATI019119





### 5.3.6 Memory Timing

Table 5-6. Read Transfer Cycle Timing

Ref.	Parameter	MIN	AX
$t_1$	Setup FBA[9:0] row address valid before RAS* active	160 ns	
$t_2$	Hold FBA[9:0] row address valid after RAS* active	22 ns	
$t_3$	Setup FBA[9:0] column address valid before CAS* active	6 ns	
$t_4$	Hold FBA[9:0] column address valid after CAS* active	22 ns	
$t_5$	Setup SBCLK falling edge (static interval) before RAS* active	86 ns	
$t_6$	Delay RAS* active to CAS* active to SBCLK active	38 ns	
$t_7$	Delay RAS* inactive to SBCLK active	86 ns	
$t_8$	Pulse Width CAS*	22 ns	
$t_9$	Pulse Width RAS[1:0]*	86 ns	
$t_{10}$	Period MCLK	16 ns	
$t_{11}$	Transition MCLK		5 ns
$t_{12}$	Setup DTE* active to RAS[1:0]* active	10 ns	
$t_{13}$	Delay RAS[1:0]* inactive to DTE* inactive		

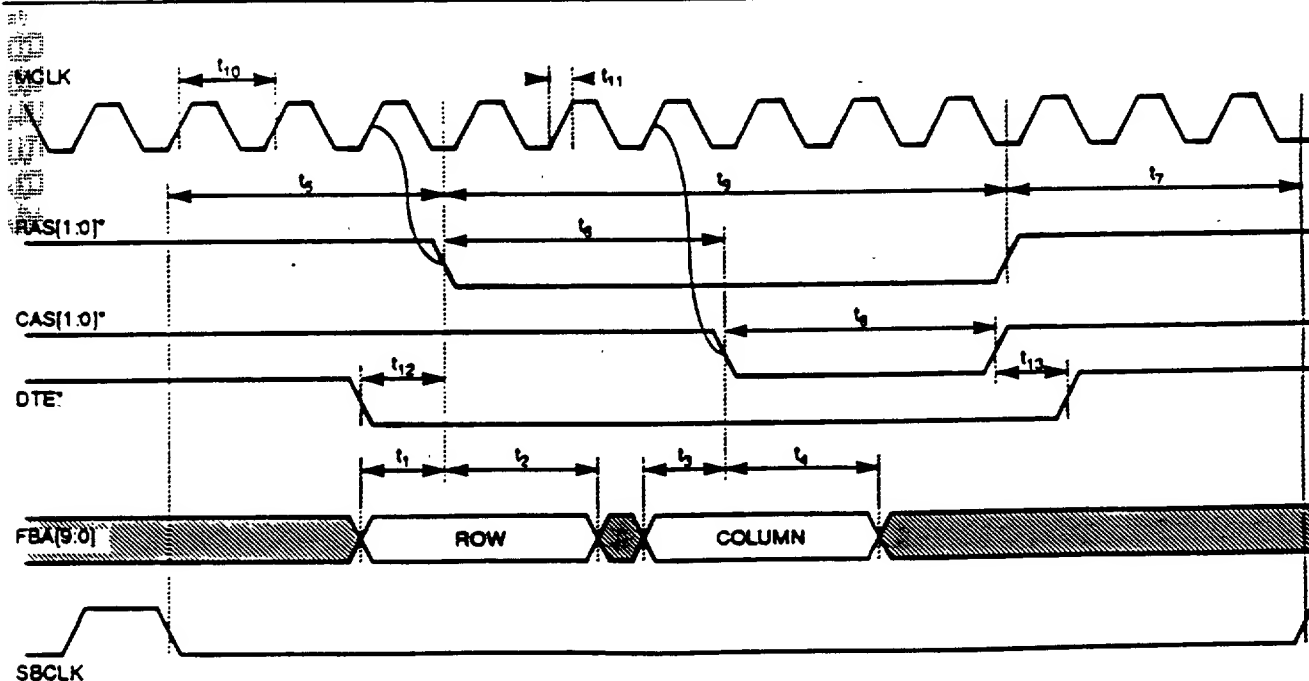


Figure 5-10. Read Transfer Cycle Timing

ATI019120



Table 5-7. CAS\* Before RAS\* Refresh Timing

Ref.	Parameter	MIN	MAX
$t_1$	Pulse Width RAS1*	86 ns	
$t_2$	Pulse Width RAS0*	86 ns	
$t_3$	Delay CAS* active to RAS1* active	38 ns	
$t_4$	Delay CAS* active to RAS0* active	38 ns	
$t_5$	Period MCLK	16 ns	
$t_6$	Transition MCLK		5 ns

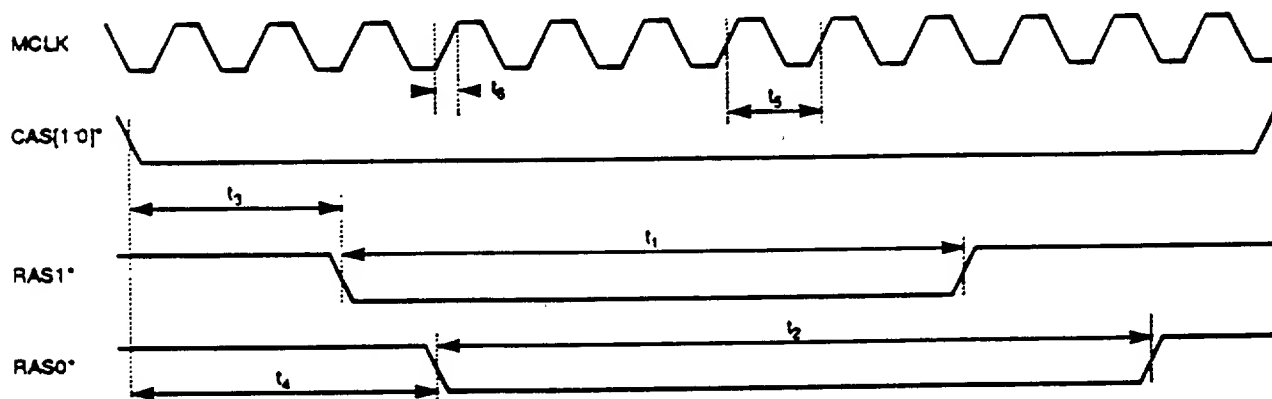


Figure 5-11. CAS\* Before RAS\* Refresh Timing

ATI019121



**Table 5-8. Memory Read and Write Timing**

Ref.	Parameter	MIN	MAX
t <sub>1</sub>	Period MCLK	16 ns	
t <sub>2</sub>	Transition MCLK		5 ns
t <sub>3</sub>	Pulse Width MCLK low	6 ns	
t <sub>4</sub>	Setup FBA[9:0] row address valid before RAS* active	6 ns	
t <sub>5</sub>	Hold FBA[9:0] row address valid after RAS* active	22 ns	
t <sub>6</sub>	Setup FBA[9:0] column address valid before CAS* active	6 ns	
t <sub>7</sub>	Hold FBA[9:0] column address valid after CAS* active	22 ns	
t <sub>8</sub>	Delay RAS* active to CAS* active	38 ns	
t <sub>9</sub>	Delay CAS* inactive to CAS* active (precharge)	11 ns	
t <sub>10</sub>	Hold RAS* active from CAS* active	38 ns	
t <sub>11</sub>	Delay RAS* inactive to RAS* active (precharge)	86 ns	
t <sub>12</sub>	Pulse Width CAS*	27 ns	
t <sub>13</sub>	Setup WE* inactive before RAS* active	38 ns	
t <sub>14</sub>	Delay FBD[31:0] valid after CAS* active (CAS* access time)	22 ns	
t <sub>15</sub>	Hold FBD[31:0] valid after CAS* inactive	0 ns	
t <sub>16</sub>	Delay FBD[31:0] valid after DTE* active	38 ns	
t <sub>17</sub>	Setup WE* active before CAS* active	6 ns	6 ns
t <sub>18</sub>	Pulse Width CAS*	27 ns	
t <sub>19</sub>	Hold WE* active from CAS* active	70 ns	
t <sub>20</sub>	Setup FBD[31:0] write valid before CAS* active	6 ns	
t <sub>21</sub>	Hold FBD[31:0] write valid after CAS* active	22 ns	

ATI019122

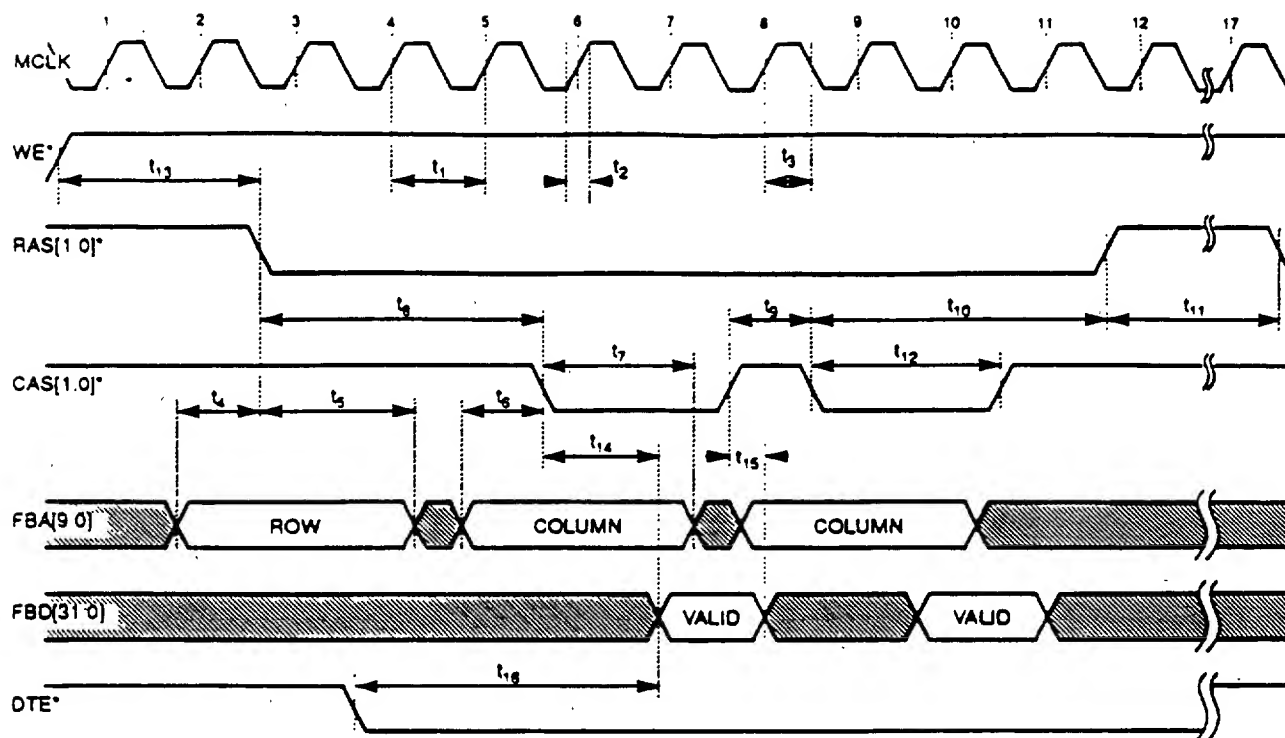


Figure 5-12. Memory Read Timing

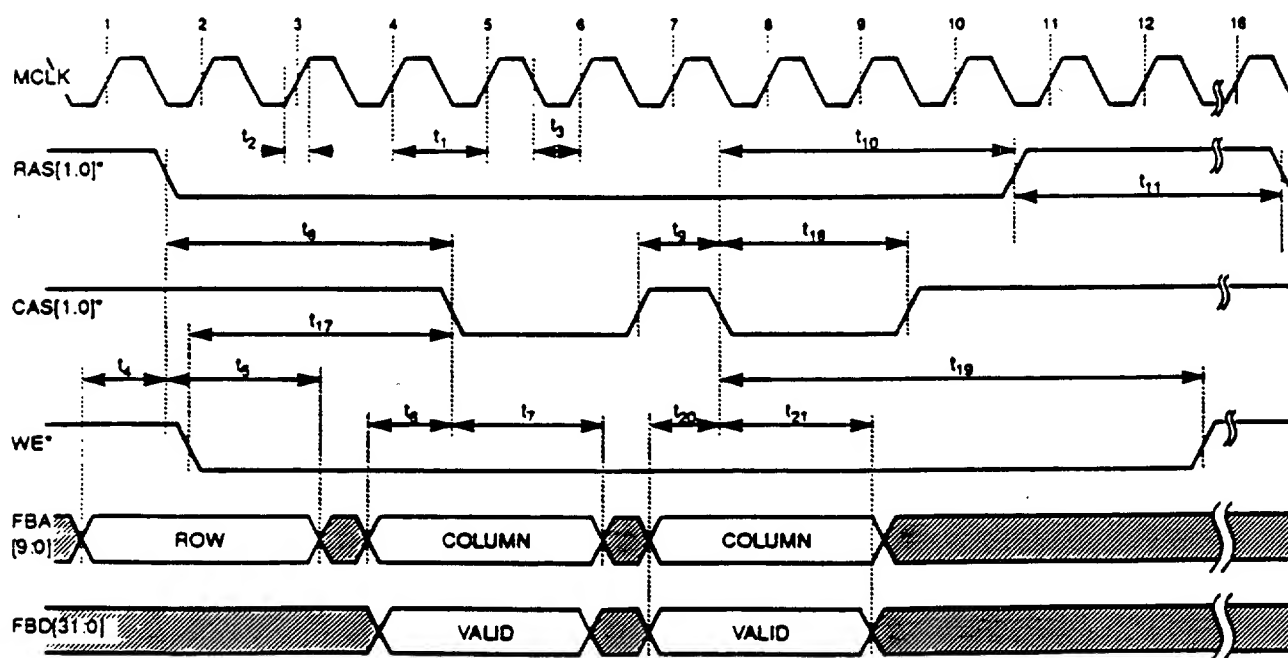


Figure 5-13. Memory Write Timing

ATI019123



## 6. PACKAGE DIMENSIONS — 160-Lead PQFP

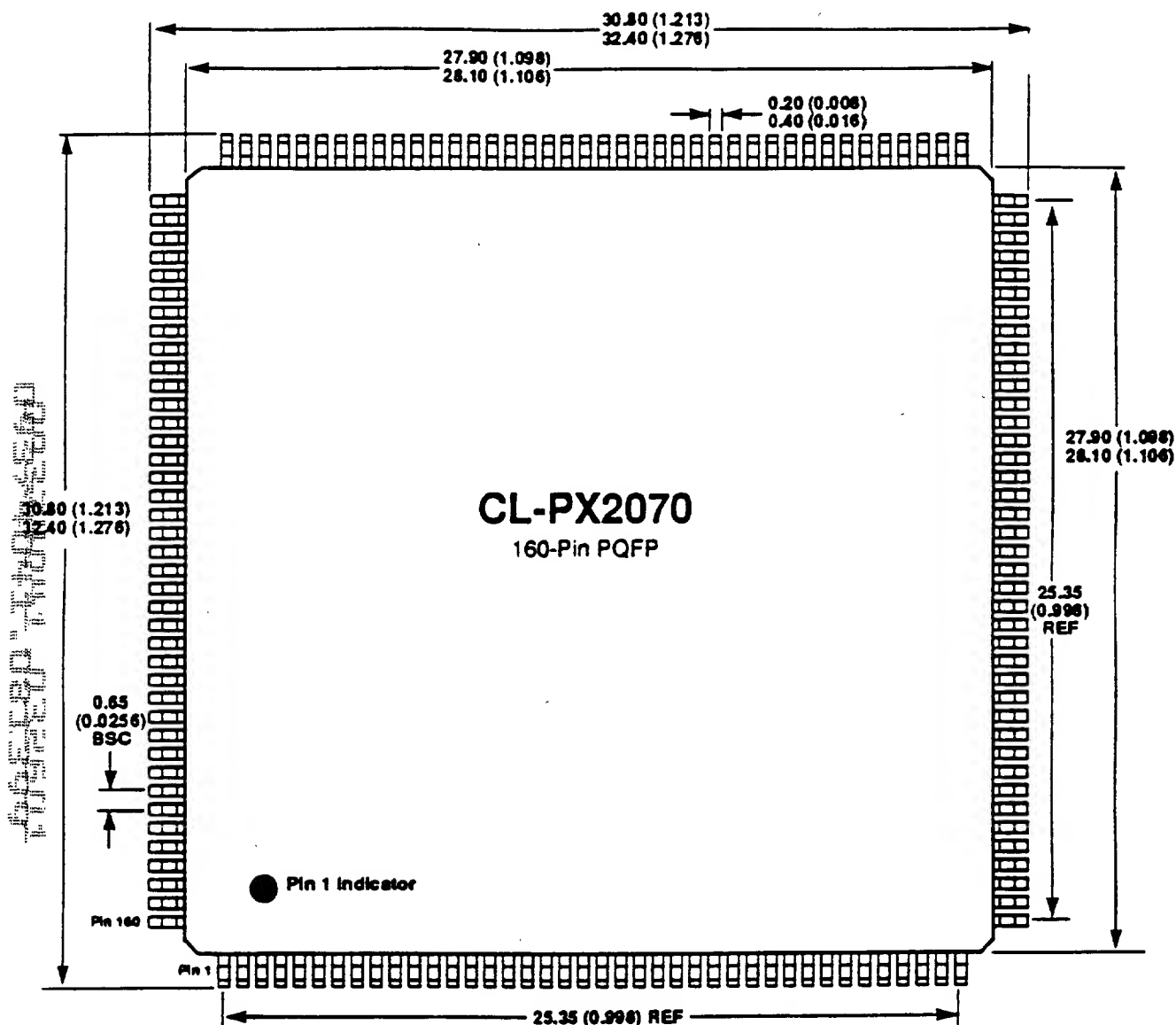


Figure 6-1. DVP Package Information

ATI019124

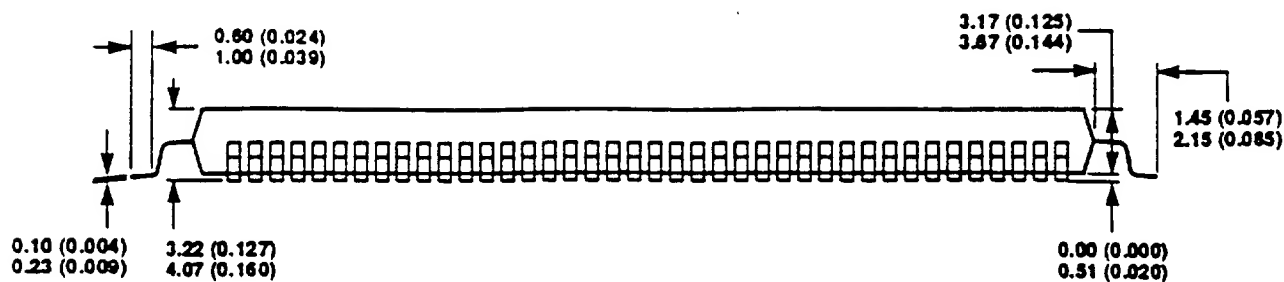
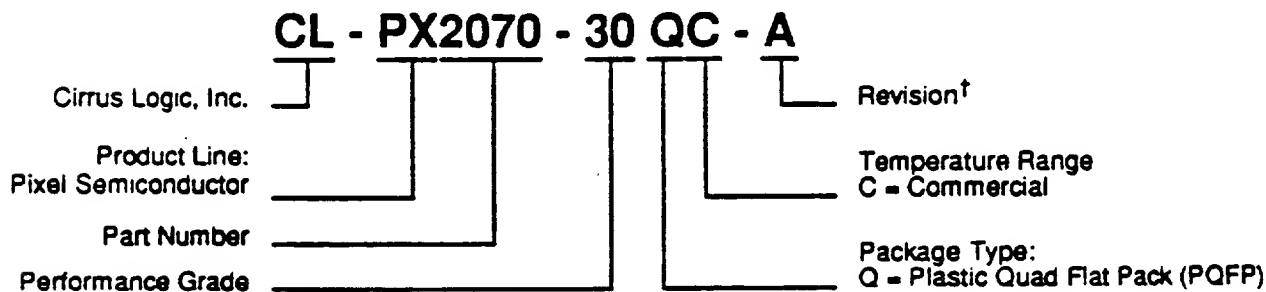


Figure 6-2. DVP Package Information (Expanded View)

## 7. ORDERING INFORMATION

When ordering the CL-PX2070 DVP, use the following format:



<sup>†</sup> Contact Cirrus Logic, Inc., for up-to-date information on revisions.

ATI019125



## APPENDIX A. DVP REGISTERS — QUICK REFERENCE

### HIU: Host Interface Unit

HIU_CSU	27C0 0290	15:12	RSVD
		11:8	VER
		7:6	RSVD
		5:3	HSB
		2	RSVD
		1	FBT
		0	PAS

HIU_OBG	27C0 0290	15:10	RSVD
		9	DRE
		8:0	RSVD

HIU_DRD	27C0 0290	15	EDT
		14:10	XC
		9:5	YC
		4:0	SIMIN

HIU_OCS	27C2 0292	15	RSVD
		14	FDNE
		13	FFNF
		12	RSVD
		11	SRC
		10	MDE
		9	DPC
		8	MPC
		7	PMC
		6:5	RSVD
		4	SR
		3:0	IEM

HIU_IRQ	27C2 0292	15:6	RSVD
		5	OBT
		4	IP2C
		3	IP1C
		2	FUN
		1	FOV
		0	WDT

HIU_RIN	27C4 0294	15	AIC
		14:0	RIN

HIU_ROT	27C6 0296	15:0	DIO
---------	--------------	------	-----

HIU_MOT	27C8 0298	15:0	MIO
---------	--------------	------	-----

HIU_ISU	0001	15:14	RSVD
		13:11	IP2S
		10:8	IP1S
		7:0	OBIS

### VBU: Video Bus Unit

VIU_MCRp	1000 1001	15	STM
		14	OFF
		13:12	OSS
		11	OVSP
		10	OHSP
		9	OBP
		8	OBT
		7	IFP
		6	ISS

5	IVSP
4	IHSP
3	IBP
2	IBT
1:0	IOM

VIU_DPCf	1002 1003	P 15:12	RSVD
		11:9	VSUDC
		8:6	IPU1DC
		5:3	IPU2DC
		2:0	ODC

VIU_WDT	1004	P 15	RSVD
		14	MMS
		13:11	MFTS
		10	WTE
		9:0	TMOUT

VIU_TEST	1006	15	MF
		14	MFID
		13:11	RSVD
		10	OBIN
		9	OVS
		8	OHS
		7	OBL
		6	OFID
		5	I2VS
		4	I2BL
		3	I2FID
		2	I1VS
		1	I1BL
		1	I2FID

VSU_HSW	1100	P 15:7	RSVD
		6:0	HSW

VSU_HAD	1101	P 15:10	RSVD
		9:0	HAD

VSU_HAP	1102	P 15:11	RSVD
		10:0	HAP

VSU_HP	1103	P 15:10	RSVD
		9:0	HP

VSU_VSW	1104	P 15:7	RSVD
		6:0	VSW

VSU_VAD	1105	P 15:10	RSVD
		9:0	VAD

VSU_VAP	1106	P 15:11	RSVD
		10:0	VAP

VSU_VP	1107	P 15	SGE
		14	SSE
		13	VFL
		12:10	RSVD
		9:0	VP

### VPU: Video Processor Unit

VPU_MCR	2000	P 15:13	RSVD
		12	ALUE
		11:8	OPFSS
		7:4	IP2FSS
		3:0	IP1FSS

IPU1_PIX	2100	15:11	RSVD
		10:0	PC

IPU1_LIC	2101	15:11	RSVD
		10:0	LC

IPU1_FLC	2102	15	RSVD
		15:0	FC

IPU1_LIR	2103	15:11	RSVD
		10:0	IRLC

IPU1_FIR	2104	15	FCE
		14:0	IRFC

IPU1_LRB	2200	15:8	RSVD
		7:0	LRB

IPU1_LRD	2201	15:8	RSVD
		7:0	LRD

IPU1_MCRf	3000 3100	P 15	FPS
		14	IM
		13	PSE
		12	CSCE
		11	LE
		10	YSP
		9:8	ODT
		7:4	OF
		3:0	IF

IPU1_XBFf	3001 3101	P 15:13	BF
		12:0	RSVD

IPU1_XBFf	3002 3102	P 15:11	RSVD
		10:0	BI

IPU1_XEIF	3003 3103	P 15:11	RSVD
		10:0	EI

IPU1_XSFI	3004 3104	P 15:5	SF
		4:0	RSVD

IPU1_XSIF	3005 3105	P 15:6	RSVD
		5:0	SI

IPU1_YBFI	3006 3106	P 15:13	BF
		12:0	RSVD

IPU1_YBFI	3007 3107	P 15:11	RSVD
		10:0	BI

IPU1_YEIF	3008 3108	P 15:11	RSVD
		10:0	EI

IPU1_YSFI	3009 3109	P 15:6	SF
		5:0	RSVD

IPU1_YSIF	300a 310a	P 15:6	RSVD
		5:0	SI

IPU1_KFCf	300b 310b	P 15:8	RSVD
		7:0	KEYFC

IPU1_MMYf	300c 310c	P 15:8	YRMAX
		7:0	YRMIN

IPU1_MMUF	300d 310d	P 15:8	UGMAX
		7:0	UGMIN

IPU1_MMYf	300e 310e	P 15:8	VBMAX
		7:0	VBMIN

ATI019126

# CL-PX2070

## Digital Video Processor



**Pixel Semiconductor**  
A Cirrus Logic Company

IPU2_PIX	2300	15:11	RSVD
		10:0	PC
IPU2_LIC	2301	14:11	RSVD
		10:0	LC
IPU2_FLC	2302	15	RSVD
		14:0	FC
IPU2_LIR	2303	15:11	RSVD
		10:0	IRLC
IPU2_FIR	2304	15	FCE
		14:0	IRFC
IPU2_MCR	3200	P 15	FPS
	3300	14	IM
		13	PSE
		12:0	RSVD
IPU2_XBI	3202	P 15:11	RSVD
	3302	10:0	BI
IPU2_XEI	3203	P 15:11	RSVD
	3303	10:0	EI
IPU2_YBI	3207	P 15:11	RSVD
	3307	10:0	BI
IPU2_YEI	3208	P 15:11	RSVD
	3308	10:0	EI
SIU_MCR	2800	15:14	RSVD
		13:12	SE
		11:10	FT
		9:5	SI2
		4:0	SI1
SIU_FCS	2801	15:14	RSVD
		13	FGF
		12	FGE
		11	FFF
		10	FFE
		9	FEF
		8	FEE
		7	FDF
		6	FDE
		5	FCF
		4	FCE
		3	FBF
		2	FBE
		1	FAF
		0	FAE
SIU_FOU	2802	15:14	RSVD
		13	FGO
		12	FGU
		11	FFO
		10	FFU
		9	FEO
		8	FEU
		7	FDO
		6	FDO
		5	FCO
		4	FCU
		3	FBO
		2	FBU
		1	FAO
		0	FAU

SIU_FAR	4001	15:7	RSVD
		6	FGR
		5	FFR
		4:0	RSVD
SIUs_SIM	2e00...	15:14	RSVD
	2e1f	13:9	OTN
		8	EP
		7:4	FA
		3:0	OBA
ALU_MCR	2900	P 15	GBM
	2901	14:13	TF
		12:9	AOP
		8:7	YOUT
		6:5	UOUT
		4:3	VOUT
		2	OPCS
		1	OPBS
		0	OPAS
ALU_TOP	2902	P 15:8	CTC
		7:0	OTC
ALU_AV	2903	P 15:8	RSVD
		7:0	AV
ALU_LOPx	2904...	P 15:0	MLOP
	2906		
ALU_CAx	2907	P 15:9	RSVD
	2908	8	TAG
	2909	7:0	CON
ALU_C8x	290a	P 15:9	RSVD
	290b	8	TAG
	290c	7:0	CON
ALU_CCx	290d	P 15:9	RSVD
	290e	8	TAG
	290f	7:0	CON
OPU_MCR	2a00	P 15	FPS
	2b00	14	IM
		13	ZE
		12:11	RSVD
		10	LSM
		9:4	RSVD
		3:0	IF
OPU_XBI	2a02	P 15:11	RSVD
	2b02	10:0	BI
OPU_XEI	2a03	P 15:11	RSVD
	2b03	10:0	EI
OPU_YBI	2a07	P 15:11	RSVD
	2b07	10:0	BI
OPU_YEI	2a08	P 15:11	RSVD
	2b08	10:0	EI
<b>RFU: Reference Frame Unit</b>			
MMU_MCR	4000	15:5	RSVD
		4	FBO
		3:0	FBC
OBUs_MCR	4800...	P 15:13	RSVD
	4870	12	LME
		11	CME

		10:6	OPM
		5	SSM
		4	YBDC
		3	XBDC
		2:0	FA
OBUs_RFX	4801...	P 15:11	RSVD
	4871	10:0	RFX
OBUs_LSL	4802...	P 15:0	LSL
	4872		
OBUs_LSH	4803...	P 15:7	RSVD
	4873	6:0	LSH
OBUs_BSY	4804...	P 15:11	RSVD
	4874	10:0	BSX
OBUs_BSY	4805...	P 15:11	RSVD
	4875	10:0	BSY
OBUs_DEC	4806...	P 15:8	RSVD
	4876	7	DM7
		6	DM6
		5	DM5
		4	DM4
		3	DM3
		2	DM2
		1	DM1
		0	DM0
DWU_MCR	4100	P 15	GCS
		14	OCC
		13	GFM
		12	GBP
		11	GBP
		10	GBP
		9	OCC
		8	IMS
		7:4	RSVD
		3	WC3
		2	WC2
		1	WC1
		0	WC0
DWU_HCR	4101	P 15:11	RSVD
		10:0	HAC
		15:8	RSVD
		7:0	MWS
DWUd_OZF	4400...	P 15:8	YZOOM
	4430	7:0	XZOOM
DWUd_RFX	4401...	P 15:11	RSVD
	4431	10:0	RFX
DWUd_LSL	4402...	P 15:0	LSL
	4432		
DWUd_LSH	4403...	P 15:7	RSVD
	4433	6:0	LSH
DWUd_WSX	4404...	P 15:11	RSVD
	4434	10:0	WSX
DWUd_WSY	4405...	P 15:11	RSVD
	4435	10:0	WSY
DWUd_DSX	4406...	P 15:12	RSVD
	4436	11:0	DSX
DWUd_DSY	4407...	P 15:12	RSVD
	4437	11:0	DSY

ATI019127





## INDEX

### Symbols

+5 VDC, signal VDD 19

### Numerics

SAD 14

16-bit I/O cycle, signal IO16 14

### A

absolute maximum ratings 84

AC characteristics/timing information 85-93

AD[15:0] 15

address

/data bus, signals AD[15:0] 15

/data bus, signals SAD[15:0] 14

bus, signals FBA[9:0] 19

column address strobes, signals CAS[1:0] 19

counter, preloading 47

enable, signal AEN 14

HIU register address map 22

I/O address map, host system 23

index, automatic increment control 27

latch, signal ADL\* 16

linear start address 76, 82

port address select 23

row address strobes, signals RAS[1:0] 19

ADL\* 16

AEN 14

Alpha Value 64

ALU

enable 44

master control 62

registers 42

ALU\_AV: Alpha Value 64

ALU\_CAx: Constant A 65

ALU\_CBx: Constant B 66

ALU\_CCx: Constant C 66

ALU\_LOPx: Logic Operation 65

ALU\_MCRf: Master Control 62

ALU\_TOP: Tag Operation 64

arithmetic operations, select 63

automatic increment control 27

### B

begin X column

fractional index 49

integer index 49, 56, 68

begin Y row

fractional index 51

integer index 51, 69

blanking

horizontal/composite

signal V1HB 18

signal V2HB 18

signal GBL 17

buffer

data, see data buffer

digital logic and interface

+5 VDC, signal VDD 19

ground, signal VSS 19

frame, see frame buffer

size 77

bus clock

serial, signal SBCKL 19

bus video data, signals V1D[15:0] 18

byte enable high, signal SBHE\* 16

### C

capacitance, input and output 84

card

data size, signal CDDS16\* 15

select feedback, signal CDSFDBK\* 15

setup, signal CDSETUP\* 15

CAS[1:0] 19

CDCHRDY 15

CDDS16\* 15

CDRESET 16

CDSETUP\* 15

timing (MCA Bus) 89

CDSFDBK\* 15

timing (MCA Bus) 89

channel

pointer, initializing 47

ready, signal CHRDY 14, 15, 16

chip select, signal CS\* 16

CHRDY 14

CHRDY\* 16

chroma key max/min 53

chrominance mask enable 74

clock

FIFO write, signal FCLK 19

memory, signal MCLK 19

pixel, signal GPCLK 17

processor, signal PCLK 17

video data, signal V1CLK 18

CL-PX2070

electrical specifications 84-93

operating mode 26

ATI019128

**CL-PX2070**  
*Digital Video Processor*

---



**Pixel  
Semiconductor**  
*A Cirrus Logic Company*

pin assignment table 11  
pin information 10-12  
signal descriptions, detailed 14-19  
CL-PX2080 79  
CMD\* 16  
Color Space Converter, enable 48  
column address strobes, signals CAS[1:0] 19  
command, signal CMD\* 16  
composite  
    /horizontal blanking  
        signal V1HB 18  
        signal V2HB 18  
    /vertical sync, signal V2VS 18  
configuration setup 23  
Constant 65, 66, 67  
control sync references, source of 32  
control tag codes 64  
CS\* 16

**D**

D[15:0] 16  
data buffer  
    direction, signal DDIR 14, 15  
    enable, signal DEN\* 14, 15  
data bus  
    signals D[15:0] 16  
    signals FBD[31:0] 19  
data transfer enable, signal DTE 19  
DDIR 14, 15  
debug  
    control 24  
    read 24  
    read enable 24  
    support registers, accessing 26  
debug enable, master 26  
DEN\* 14, 15  
detailed signal descriptions 14-19  
diagnostic  
    information, accessing 24  
    mode, controlling 24  
display  
    start 83  
    window, posting operation control 26  
    zoom factor 81  
DTE 19  
DWU  
    master control 78  
    register posting mode 26  
    registers 72  
        DWU\_DSa: Display Start 83  
        DWU\_DZF: Display Zoom Factor 81  
        DWU\_HAC: Horizontal Active Count 80

DWU\_LSb: Linear Start Address 82  
DWU\_MCR: Master Control 78  
DWU\_RFX: Reference Frame X Size 81  
DWU\_WSa: Window Size 82

**E**

electrical specifications 84-93  
enable input  
    signal V1IEN\* 18  
    signal V2IEN\* 18  
end 51  
    X column integer index 57, 68  
    Y row integer index 58, 69  
error detection trap 24  
exit point 61

**F**

FBA[9:0] 19  
FBD[31:0] 19  
FCLK 19  
FDRY\* 19  
field  
    count 46, 54  
    interrupt request 46, 55  
    polarity select 48, 56, 67  
    toggle 58  
Field Lock 74  
FIFO  
    association 61, 75  
    flag  
        empty 59  
        full 59  
        overflow 60  
        underflow 60  
    overflow 25, 60  
    ready, signal FDRY\* 19  
    underflow 25  
    write clock, signal FCLK 19  
FL 74  
frame buffer  
    configuration 73  
    depth 73  
    interface  
        detailed signal descriptions 19  
        pin assignments 13  
        signals 19  
    type 23  
Functional Signal Groups 11

ATI019129



## G

GBL 17  
GHS 17  
GPCLK 17  
graphics  
    blank polarity 79  
    field  
        mode 79  
        polarity 79  
    hsync polarity 79  
    overlay interface  
        detailed signal descriptions 17  
        pin assignments 13  
    vsync polarity 79  
ground, signal VSS 19  
GVS 17

## H

hardware configuration data, storing 23  
HIU  
    register address map 22  
    registers 22-29  
        HIU\_CSU: Configuration Setup 23  
        HIU\_DBG: Debug Control 24  
        HIU\_DRD: Debug Read 24  
            access to 24  
        HIU\_IRQ: Interrupt Request 25  
        HIU\_MDT: Memory Data Port 28  
        HIU\_OCS: Operation Control/Status 26  
        HIU\_RDT: Register Data Port 28  
        HIU\_RIN: Register Index 27  
    registers accessed by the Register Data Port 22  
horizontal  
    active  
        count 80  
        delay 36  
        pixels 36  
    period 36  
    sync  
        signal GHS 17  
        signal V1HS 18  
        signal V2CLK 18  
        signal V2HS 18  
        width 35  
horizontal/composite blanking  
    signal V1HB 18  
    signal V2HB 18  
host system  
    bus 23  
    specifying 23

## I

I/O address map, used by host system 23  
I/O read  
    signal IOR\* 14, 16  
    timing, local hardware interface 91  
I/O write  
    signal IOW\* 14, 16  
    timing  
        ISA bus 86  
        local hardware interface 90  
index  
    address, automatic increment control 27  
    value, specifying 27  
input  
    data format 49, 67  
    enable  
        signal V1IEN\* 18  
        signal V2IEN\* 18  
    stream, specifying the format of 49  
    tag multiplexer, controlling the input selection 48  
    VIU input/output mode 31  
    voltages 84  
interface mode 56, 67  
    select 79  
    specifying interlaced or non-interlaced 48  
internal sync generator, controls for 38  
interrupt  
    enable, mask 27  
    mode 29  
    request 25  
        field count 47, 55  
        line count 46, 55  
        signal IRQ 14, 15, 16  
IO16\* 14  
IOR\* 14, 16  
IOW\* 14, 16  
IPU1 49, 51, 52  
    counter 25  
    datapath control 32  
    field sync select 44  
    interrupt select 29  
    master control 48  
    registers 39  
        IPU1\_FIR: Field Count Interrupt Request 46  
        IPU1\_FLC: Field Count 46  
        IPU1\_KFCf: Key Function Code 52  
        IPU1\_LIC: Line Count 45  
        IPU1\_LRB: LUT RAM Base Address 47  
        IPU1\_LRD: LUT RAM Data 47  
        IPU1\_MCRf: Master Control 48  
        IPU1\_MMxf: Chroma Key Max/Min 53  
        IPU1\_PIX: Pixel Count 45

ATI019130



IPU1\_XBnf: X Begin 49  
IPU1\_XEIf: X End 50  
IPU1\_XSI2 50  
IPU1\_XSnf: X Shrink 50  
IPU1\_YBnf: Y Begin 51  
IPU1\_YEIf: Y End 51  
IPU1\_YSnf: Y Shrink 52

**IPU2**  
counter 25  
datapath control 32  
field sync select 44  
interrupt select 29  
registers 40  
IPU2\_FIR: Field Count Interrupt Request 55  
IPU2\_FLC: Field Count 54  
IPU2\_LIC: Line Count 54  
IPU2\_LIR: Line Count Interrupt Request 55  
IPU2\_PIX: Pixel Count 54  
IPU2\_XBIf: X Begin 56  
IPU2\_XEIf: X End 57  
IPU2\_YBIf: Y Begin 57  
IPU2\_YEIf: Y End 58

**IRQ** 14, 15, 16

**ISA bus**  
detailed signal descriptions 14  
I/O write timing 86  
pin assignments 12  
processor interface signals 14

## K

key function code 52  
key maximum/minimum 53

## L

lead temperature 84  
leakage 84  
line count 45, 54  
current field 45, 54  
interrupt request 55  
to interrupt request 46  
line start mode 67  
linear start address 76, 82  
local hardware interface  
detailed signal descriptions 16  
I/O read timing 91  
I/O write timing 90  
pin assignments 12  
processor interface signals 16  
logic operation 65  
Luminance Mask Enable 74  
LUT enable 48

**LUT RAM**  
base address 47  
data 47

## M

**MIO\*** 15  
manual mode start 33  
master control  
ALU 62  
DWU 78  
IPU1 48  
IPU2 56  
MMU 73  
OBU 74  
OPU 67  
SIU 58  
VIU 30  
VPU 44  
maximum ratings, absolute 84  
**MCA bus**  
CDSETUP\* timing 89  
CDSFDBK\* timing 89  
detailed signal descriptions 15  
pin assignments 12  
processor interface signals 15  
timing, write cycle 88  
**MCLK** 19  
memory  
clock, signal MCLK 19  
data 28  
or I/O cycle, signal MIO\* 15  
Minimum Window Separation 80  
**MMU**  
master control 73  
registers 70  
MMU\_MCR: Master Control 73  
mode, stall 31  
multiplexers  
key function code 52  
logical operation 65  
logical, specifying the constant values for 65

## O

object buffer  
association 61  
termination 25, 29  
**OBU** ATI019131  
master control 74  
registers 70  
OBU\_BSa: Buffer Size 77  
OBU\_DEC: Decimate Control 78



OBU\_LSB: Linear Start Address 76  
 OBU\_MCR: Master Control 74  
 OBU\_RFX: Reference Frame X Size 75  
 Occluded Window Control 79  
 Offset to Next Instruction 61  
 operands  
   3-operand bit mask 62  
   source select 63  
   specifying constant values for 65, 66  
 operating mode 26, 74  
 operations, control/status 26  
 OPU  
   datapath control 33  
   field sync select 44  
   master control 67  
   registers 43  
     OPU\_MCRf: Master Control 67  
     OPU\_XBIf: X Begin 68  
     OPU\_XEIf: X End 68  
     OPU\_YBIf: Y Begin 69  
     OPU\_YEIf: Y End 69  
 ordering information 99  
 output  
   data 48  
   stream, specifying the format of 48  
   tag codes 64  
 output voltages 84  
 overflow condition, FIFO 25  
   flag 60  
 package dimensions 98  
 PCLK 17  
 phase  
   signal V1PH 18  
   signal V2PH 18  
 pin assignment table 11  
 pin diagram 10  
 pin information 10-12  
 pixel clock, signal GPCLK 17  
 pixel count 45, 54  
   current line 45, 54  
 Plastic Quad Flat Pack, 160-lead 10  
 polarity, graphics 79  
 port address select 23  
 posting control, master 26  
 posting mode control 26  
 power and ground  
   detailed signal descriptions 19  
   signals 19  
 power interface, pin assignments 13  
 power supply voltage 84

PQFP 10  
 prescaler enable 48, 56  
 processor clock, signal PCLK 17  
 processor interface  
   ISA bus mode  
     detailed signal descriptions 14  
     pin assignments 12  
   local hardware interface mode  
     detailed signal descriptions 16  
     pin assignments 12  
   MCA bus mode  
     detailed signal descriptions 15  
     pin assignments 12

## R

RAS[1:0] 19  
 Reference Frame, X size 75, 81  
 register  
   access, during a read cycle 26  
   address map, HIU 22  
   data port 28  
   I/O data 28  
   index 27  
   posting 26  
   select, signals RS[2:1] 16  
 registers  
   HIU 22-29  
   RFU 70-83  
   VBU 30-38  
   VPU 39-69  
 reset  
   signal RESET 14, 16, 17  
   soft 27  
 RFU registers 70-83  
 row address strobes, signals RAS[1:0] 19  
 RS[2:1] 16

## S

S0\* 15  
 S1\* 15  
 SBCLK 19  
 SBHE\* 16  
 sequencer enable 58  
 Sequencer Instruction Memory 61  
   current index 24  
 serial bus clock, signal SBCLK 19  
 serial port output enable, signals SOE[1:0]\* 19  
 signal descriptions, detailed 14-19  
   frame buffer interface 19  
   graphics overlay interface 17  
   power and ground 19

ATI019132



processor interface  
  ISA bus mode 14  
  local hardware interface mode 16  
  MCA bus mode 15  
Video 18  
single sweep mode 75  
SIU  
  master control 58  
  registers 41  
    master control 58  
    SIU\_FARC: FIFO Auto Reset Control 62  
    SIU\_FCS: FIFO Control/Status 59  
    SIU\_FOU: FIFO Overflow/Underflow 60  
    SIU\_MCR: Master Control 58  
    SIU\_SIM: Sequencer Instruction Memory 61  
SOE[1:0]\* 19  
soft reset 27  
specifications, electric 84-93  
stall  
  mode 31  
  request, signal STALLRQ\* 18  
STALL\* 18  
STALLRQ\* 18  
Start Index 58  
status  
  read select 26  
storage temperature 84  
strokes  
  column address, signal CAS[1:0] 19  
  row address, signals RAS[1:0] 19  
supply currents, digital and total 84  
sync, horizontal  
  signal GHS 17  
  signal V1HS 18  
  signal V2CLK 18  
  signal V2HS 18  
sync, vertical  
  signal GVS 17  
sync, vertical/composite  
  signal V1VS 18  
  signal V2VS 18

## T

tag 65, 66  
  format 63  
  operation 64  
timeout condition, watchdog timer 25  
timing  
  CDSETUP\* (MCA bus) 89  
  CDSFDBK\* (MCA bus) 89  
  I/O read 91  
  I/O write 86, 90

information/AC characteristics 85-93  
write cycle (MCA bus) 88

## U

U output Source Select 63  
underflow condition, FIFO 25  
  flag 60

## V

V output Source Select 63  
V1CLK 18  
V1D[15:0] 18  
V1HB 18  
V1HS 18  
V1IEN\* 18  
V1PH 18  
V1VS 18  
V2CLK 18  
V2D[15:0] 18  
V2HB 18  
V2HS 18  
V2IEN 18  
V2PH 18  
V2VS 18  
VBU registers 30-38  
+5 VDC, signal VDD 19  
VDD 19  
vertical active  
  delay 37  
  pixels 38  
vertical period 38  
vertical sync  
  signal GVS 17  
  timing 35  
  width 37  
vertical/composite sync  
  signal V1Vs 18  
  signal V2VS 18  
video  
  interface signals 18  
  port interfaces, characteristics 30  
  processor, device version 23  
video data bus  
  signals V1D[15:0] 18  
  signals V2D[15:0] 18  
video data clock, signal V1CLK 18  
video input 31  
video output 31  
  field polarity 31  
video port 1  
  detailed signal descriptions 18

ATI019133



pin assignments 13  
VIU  
  registers  
    VIU\_DPCf: Datapath Control 32  
    VIU\_MCRp: Master Control 30  
voltages, low and high 84  
VPU  
  global control 39, 44  
  master control 44  
  registers 39-69  
  VPU\_MCR: Master Control 44

VSS 19

VSU registers

VSU\_HAD: Horizontal Active Delay 36  
VSU\_HAP: Horizontal Active Pixels 36  
VSU\_HP: Horizontal Period 36  
VSU\_HSW: Horizontal Sync Width 35  
VSU\_VAD: Vertical Active Delay 37  
VSU\_VAP: Vertical Active Pixels 38  
VSU\_VP: Vertical Period 38  
VSU\_VSW: Vertical Sync Width 37  
VSU sync timing 35

## W

watchdog timer, signal IRQ 25

WE\* 19

windows

  aligning between pixels 49  
  controls 79  
  size 82

write

  cycle timing (MCA Bus) 88  
  decimation, mask bits 78  
  enable, signal WE\* 19

## X

X 50

X Begin 49, 56

X BLT, direction control 75

X counter 24

X display start 83

X End 50, 57

X prescaler, enabling and disabling 48

X shrink 50

X window size 83

X zoom factor 81

## Y

Y 52

Y Begin 51, 57

Y BLT direction control 75

Y counter 24

Y display start 83

Y End 51, 58, 69

Y output source select 63

Y row integer index 51

Y scaling path, enabling and disabling 48

Y shrink 52

Y window size 83

Y zoom factor 81

## Z

ZC[3:0] 17

zoom

  control bus, signals ZC[3:0] 17  
  enable 67



## INDEX OF CONTROL REGISTERS

### A

ALU\_AV: Alpha Value 64  
ALU\_CAx: Constant A 65  
ALU\_CBx: Constant B 66  
ALU\_CCx: Constant C 66  
ALU\_LOPx: Logic Operation 65  
ALU\_MCRf: ALU Master Control 62  
ALU\_TOP: Tag Operation 64

### C

Chroma Key Max/Min 53  
Configuration Setup 22, 23  
Constant A 65  
Constant B 66  
Constant C 66

### D

Datapath Control 32  
Debug  
    Control 24  
    Control/Status 22  
    Read 22, 24  
Display Window  
    Display Zoom Factor 81  
    Horizontal Control 80  
    Linear Start Address 82  
    Master Control 78  
    Reference Frame Size 81  
    Size 82  
    Start 83  
DWU\_HCR: Display Window Horizontal Control Register 80  
DWU\_MCR: Display Window Master Control 78  
DWUd\_DSa: Display Window Start 83  
DWUd\_DZF: Display Window Display Zoom Factor 81  
DWUd\_LSa: Display Window Linear Start Address 82  
DWUd\_RFX: Display Window Reference Frame Size 81  
DWUd\_WSa: Display Window Size 82

### F

Field Count 46, 54  
    Interrupt Request 46, 55  
FIFO  
    auto reset control 62  
    Control Status 59  
    Overflow/Underflow 60

### H

HIU\_0 22  
HIU\_1 22  
HIU\_2 22  
HIU\_3 22  
HIU\_4 22  
HIU\_CSU: Configuration Setup 22, 23  
HIU\_DBG: Debug Control 24  
HIU\_DBG: Debug Control/Status 22  
HIU\_DRD: Debug Read 22, 24  
HIU\_IRQ: Interrupt Request 22, 25  
HIU\_ISU: Interrupt Select 22  
HIU\_ISU: Interrupt Setup 29  
HIU\_MDT: Memory Data Port 22, 28  
HIU\_OCS: Operation Control/Status 22, 26  
HIU\_RDT: Register Data Port 22, 28  
HIU\_RIN: Register Index 22, 27  
Horizontal  
    Active Delay 36  
    Active Period 36  
    Period 36  
    Sync Width 35

### I

Interrupt  
    Request 22, 25  
    Select 22  
    Setup 29  
IPU1\_FIR: Field Count Interrupt Request 46  
IPU1\_FLC: Field Count 46  
IPU1\_KFCf: Key Function Code 52  
IPU1\_LIC: Line Count 45  
IPU1\_LIR: Line Count Interrupt Request 46  
IPU1\_LRB: LUT RAM Base Address 47  
IPU1\_LRD: LUT RAM Data 47  
IPU1\_MCRf: IPU1 Master Control 48  
IPU1\_MMxf: Chroma Key Max/Min 53  
IPU1\_PIX: Pixel Count 45  
IPU1\_XBnf: X Begin 49  
IPU1\_XEtf: X End 50  
IPU1\_XSnt: X Shrink 50  
IPU1\_YBnf: Y Begin 51  
IPU1\_YEtf: Y End 51  
IPU1\_YSnt: Y Shrink 52  
IPU2\_FIR: Field Count Interrupt Request 55  
IPU2\_FLC: Field Count 54  
IPU2\_LIC: Line Count 54  
IPU2\_LIR: Line Count Interrupt Request 55  
IPU2\_MCRf: IPU2 Master Control 56





IPU2\_PIX: Pixel Count 54  
IPU2\_XBIf: X Begin 56  
IPU2\_XEIf: X End 57  
IPU2\_YBIf: Y Begin 57  
IPU2\_YEIf: Y End 58

## K

Key Function Code 52

## L

Line Count 45, 54  
    Interrupt Request 46, 55  
Logic Operation 65  
LUT RAM  
    Base Address 47  
    Data 47

## M

Memory Data Port 22, 28  
MMU\_MCR: MMU Master Control 73

## O

Object Buffer  
    Decimate Control 78  
    Linear Start Address 76  
    Master Control 74  
    Reference Frame Size 75  
    Size 77  
OBUo\_BSa: Object Buffer Size 77  
OBUo\_DEC: Object Buffer Decimate Control 78  
OBUo\_LSB: Object Buffer Linear Start Address 76  
OBUo\_MCR: Object Buffer Master Control 74  
OBUo\_RFX: Object Buffer Reference Frame Size 75  
Operation Control/Status 22, 26  
OPU\_MCRf: OPU Master Control 67  
OPU\_XBIf: X Begin 68  
OPU\_XEIf: X End 68  
OPU\_YBIf: Y Begin 69  
OPU\_YEIf: Y End 69

## P

Pixel Count 45, 54

## R

Register  
    Data Port 22, 28  
    Index 22, 27

## S

Sequencer Instruction Memory 61  
SIU\_FARC: FIFO Auto Reset Control 62  
SIU\_FCS: FIFO Control Status 59  
SIU\_FOU: FIFO Overflow/Underflow 60  
SIU\_MCR: SIU Master Control 58  
SIUs\_SIM: Sequencer Instruction Memory 61

## T

Tag Operation 64  
test register 34

## V

Vertical  
    Active Delay 37  
    Active Period 38  
    Period 38  
    Sync Width 37  
VIU\_DPCf: Datapath Control 32  
VIU\_MCRp: VIU Master Control 30  
VIU\_TEST: test register 34  
VIU\_WDT: Watchdog Timer 33  
VPU\_MCR: VPU Master Control 44  
VSU\_HAD: Horizontal Active Delay 36  
VSU\_HAP: Horizontal Active Period 36  
VSU\_HP: Horizontal Period 36  
VSU\_HSW: Horizontal Sync Width 35  
VSU\_VAD: Vertical Active Delay 37  
VSU\_VAP: Vertical Active Period 38  
VSU\_VP: Vertical Period 38  
VSU\_VSW: Vertical Sync Width 37

## W

Watchdog Timer 33

## X

X Begin 49, 56, 68  
X End 50, 57  
X Shrink 50

## Y

Y Begin 51, 57, 69  
Y End 51, 58, 69  
Y Shrink 52

ATI019136



## INDEX OF REGISTER FIELDS

### A

AIC: Automatic Increment Control 27  
ALUE: ALU Enable 44  
AOP: Arithmetic Operation Select 63  
AV: Alpha Value 64

### B

BF: Begin X Column Fractional Index 49  
BF: Begin Y Row Fractional Index 51  
BI: Begin X Column Integer Index 49, 56, 68  
BI: Begin Y Row Integer Index 51, 57, 69  
BSX: Buffer X Size 77  
BSY: Buffer Y Size 77

### C

CME: Chrominance Mask Enable 74  
CON: Constant 65, 66, 67  
CSCE: Color Space Converter Enable 48  
CTC: Control Tag Code 64

### D

Datapath Control 32, 33  
DIO: Register Data I/O 28  
DM7:DM0 78  
DMAW: DMA Wait State 26  
DMD: DMA Direction 26  
DPC: Display Window Posting Operation Control 26  
DRE: Debug Read Enable 24  
DSX: Display X Start 83  
DSY: Display Y Start 83

### E

EDT: Error Detection Trap 24  
EI: End X Column Integer Index 50, 57, 68  
EI: End Y Row Integer Index 51, 58, 69  
EP: Exit Point 61

### F

FA: FIFO Association 61, 75  
FBC: Frame Buffer Address Configuration 73  
FBD: Frame Buffer Data Bus Width 73  
FBT: Frame Buffer Jumper State 23  
FC: Field Count 46, 54  
FCE: Field Count Enable 47, 55  
FDH: FIFO D Half Full 26  
FFH: FIFO F Half Full 26

FGE - FAE: FIFO Empty Flags 59  
FGF - FAF: FIFO Full Flags 59  
FGO - FAO: FIFO Overflow Flags 60  
FGU - FAU: FIFO Underflow Flags 60  
Field Sync Select 44  
FL: Field Lock 74  
FOV: FIFO Overflow 25  
FPS: Field Polarity Select 48, 56, 67  
FT: Field Toggle 58  
FUN: FIFO Underflow 25

### G

GBM: Three-Operand Bit Mask Selecting Tag Source 62  
GBP: Graphics Blank Polarity 79  
GCS: Graphics Clock Select 79  
GFM: Interface Mode Select 79  
GFP: Graphics Field Polarity 79  
GHSP: Graphics Hsync Polarity 79  
Graphics Field Mode 79  
GVSP: Graphics Vsync Polarity 79

### H

HAC: Horizontal Active Count 80  
HAD: Horizontal Active Delay 36  
HAP: Horizontal Active Pixels 36  
HP: Horizontal Period 37  
HSB: Host System Bus 23  
HSW: Horizontal Sync Width 35

### I

I1BL 34  
I1FID 34  
I1VS 34  
I2BL 34  
I2FID 34  
I2VS 34  
IBP: Input Video Blank Polarity 31  
IBT: Input Video Blank Type 31  
IEM: Interrupt Enable Mask 27  
IF: Input Data Format 49, 67  
IFP: Input Video Field Polarity 31  
IHSP: Input Video Horizontal Sync Polarity 31  
IM: Interface Mode 48, 56, 67  
IMS: Graphics Field Mode 79  
Interface Mode Select 79  
INTR 34  
IOM: Input/Output Mode 31

ATI019137



IP1C: IPU1 Counter 25  
IP1FSS: IPU1 Field Sync Select 44  
IP1S: IPU1 Select 29  
IP2C: IPU2 Counter 25  
IP2FSS: IPU2 Field Sync Select 44  
IP2S: IPU2 Interrupt Select 29  
IPU1DC: IPU1 Datapath Control 32  
IPU2DC: IPU2 Datapath Control 32  
IRFC: Interrupt Request Field Count 47, 55  
IRLC: Interrupt Request Line Count 46, 55  
ISS: Input Sync Source 31  
IVSP: Input Video Vsync Polarity 31

## K

KEYFC: Key Function Code 52

## L

LC: Line Count 45  
LC: Line Count Current Field 54  
LE: LUT Enable 48  
LME: Luminance Mask Enable 74  
LRB: LUT RAM Base Address 47  
LRD: LUT RAM Data 47  
LSH: Linear Start Address High 76, 82  
LSL: Linear Start Address Low 76, 82  
LSM: Line Start Mode 67

## M

MDE: Master Debug Enable 26  
MF 34  
MFD 34  
MFTS: Master Field Toggle Select 33  
MIO: Memory Data I/O 28  
MLOP: Multiplexor Logical Operation 65  
MMS: Manual Mode Start 33  
MPC: Master Posting Control 26  
MWS: Minimum Window Separation 80

## O

OBA: Object Buffer Association 61  
OBIN 34  
OBIS: Object Buffer Termination Interrupt Request 29  
OBL 34  
OBP: Output Video Blank Polarity 31  
OBT: Object Buffer Termination 25  
OBT: Output Video Blank Type 31  
OCC: Occluded Window Control 79  
ODC: OPU Datapath Control 33  
ODT: Output Data Tag 48  
OF: Output Data Format 48

OFID 34  
OFP: Output Video Field Polarity 31  
OHS 34  
OHSP: Output Video Horizontal Sync Polarity 31  
OPAS: Operand A Source Select 63  
OPBS: Operand B Source Select 63  
OPCS: Operand C Source Select 63  
OPFSS: OPU Field Sync Select 44  
OPM: Operation Mode 74  
ORS: Operation Restart Select 24  
OSS: Output Video Sync Source 31  
OTC: Output Tag Code 64  
OTN: Offset to Next Instruction 61  
OVS 34  
OVSP: Output Video Vertical Sync Polarity 31

## P

PAS: Port Address Select 23  
PC: Pixel Count 45  
PC: Pixel Count Current Line 54  
PMC: Posting Mode Control 26  
PSE: Prescaler Enable 48, 56

## R

RFX: Reference Frame X Size 75  
RFX: Reference Frame X size 81  
RIN: Register Index 27

## S

SE: Sequencer Enable 58  
SF: X Shrink Fraction 50  
SF: Y Shrink Fraction 52  
SGE: Sync Generator Enable 38  
SI: X Shrink Integer 50  
SI: Y Shrink Integer 52  
SI1: Start Index 1 58  
SI2: Start Index 2 58  
SIMIN: Sequence Instruction Memory Current Index 24  
SR: Soft Reset 27  
SRC: Status Read Select 26  
SSE: Single Sweep Enable 38  
SSM: Single Sweep Mode 75  
STM: Stall Mode 31

## T

TAG: Constant Tag Bit 65, 66  
TF: Tag Format 63  
TMOUT: Timeout 33

ATI019138

**CL-PX2070**  
*Digital Video Processor*

---



**Pixel  
Semiconductor**  
*A Cirrus Logic Company*

**U**

UGMAX: Key U/G Maximum 53  
UGMIN: Key U/G Minimum 53  
UOUT: U Output Source Select 63

**V**

VAD: Vertical Active Delay 37  
VAP: Vertical Active Pixels 38  
VBMAX: Key V/B Maximum 53  
VBMIN: Key V/B Minimum 53  
VER: Video Processor Device Version 23  
VOUT: V Output Source Select 63  
VP: Vertical Active Count 38  
VSUDC: VSU Datapath Control 32  
VSW: Vertical Sync Width 37

**W**

WC3 - WC0: Window Controls 79  
WDT: Watchdog Timer 25  
Write Decimation Mask Bits 7:0 78  
WSX: Window X Size 83  
WSY: Window Y Size 83  
WTE: Watchdog Timer Enable 33

**X**

XBDC: X BLT Direction Control 75  
XC: X Counter 24  
XZOOM: X Zoom Factor — pixel replication value 81

**Y**

YBDC: Y BLT Direction Control 75  
YC: Y Counter 24  
YOUT: Y Output Source Select 63  
YRMAX: Key Y/R Maximum 53  
YRMIN: Key Y/R Minimum 53  
YSP: Y Scaling Path 48  
YZOOM: Y Zoom Factor — line replication value 81

**Z**

ZE: Zoom Enable 67

ATI019139



**CL-PX2070**  
**Digital Video Processor**

## NOTES

[illegible]

ATI019140

## CL-PX2070

### Digital Video Processor



## NOTES

**SECRET**

ATI019141



**Pixel  
Semiconductor**  
A Cirrus Logic Company

**CL-PX2070**  
*Preliminary Data Book*

---

## **Cirrus Logic Direct Sales Offices**

### **Domestic**

#### **N. CALIFORNIA**

San Jose  
TEL: 408/436-7110  
FAX: 408/437-8960

#### **S. CALIFORNIA**

Tustin  
TEL: 714/258-8303  
FAX: 714/258-8307

Thousand Oaks  
TEL: 805/371-5381  
FAX: 805/371-5382

#### **ROCKY MOUNTAIN AREA**

Denver, CO  
TEL: 303/768-9696  
FAX: 303/768-9695

#### **SOUTH CENTRAL AREA**

Austin, TX  
TEL: 512/794-8490  
FAX: 512/794-8069

Plano, TX  
TEL: 214/985-2334  
FAX: 214/964-3119

#### **NORTHEASTERN AREA**

Andover, MA  
TEL: 508/474-9300  
FAX: 508/474-9149

Philadelphia, PA  
TEL: 215/625-0781  
FAX: 215/625-0731

#### **CENTRAL AREA**

Chicago, IL  
TEL: 708/490-5940  
FAX: 708/490-5942

#### **SOUTH EASTERN AREA**

Boca Raton, FL  
TEL: 407/362-5225  
FAX: 407/362-5235

### **International**

#### **GERMANY**

Herrsching  
TEL: 49/08152-2030  
FAX: 49/08152-6211

#### **JAPAN**

Tokyo  
TEL: 81/3-3340-9111  
FAX: 81/3-3340-9120

#### **SINGAPORE**

TEL: 65/3532122  
FAX: 65/3532166

#### **TAIWAN**

Taipei  
TEL: 886/2-718-4533  
FAX: 886/2-718-4526

#### **UNITED KINGDOM**

Hertfordshire, England  
TEL: 44/0727-872424  
FAX: 44/0727-875919

---

## **The Company**

Cirrus Logic, Inc., is a leading supplier of high-integration peripheral controller circuits for mass storage, graphics, and data communications. The company also produces state-of-the-art software and firmware to complement its product lines. Cirrus Logic technology is used in leading-edge personal computers, engineering workstations, and office automation.

Pixel Semiconductor, Inc., a subsidiary of Cirrus Logic, Inc., is a developer of integrated circuits for advanced display systems. These circuits enable the integration of real-time video with traditional computer graphics.

Cirrus Logic's fabless manufacturing strategy, unique in the semiconductor industry, employs a full manufacturing infrastructure to ensure maximum product quality, availability and value for our customers.

Talk to our systems and applications specialists; see how you can benefit from a new kind of semiconductor company.

© Copyright Pixel Semiconductor, Inc., 1993

---

*Preliminary product information describes products which are in production, but for which full characterization data is not yet available. Pixel Semiconductor, Inc., believes that the information contained in this document is accurate and reliable. However, it is marked Preliminary and is subject to change without notice. Pixel Semiconductor, Inc., assumes no responsibility for its use, nor for infringements of patents or other rights of third parties. This document implies no license under patents or copyrights. Trademarks in this document belong to their respective companies.*

---

CIRRUS LOGIC, Inc.,

3100 West Warren Ave.  
TEL: 510/623-8300

Fremont, CA 94538  
FAX: 510/226-2180

742070-002

ATI019142



Pixel  
Semiconductor  
A Cirrus Logic Company

# CL-PX2080

Preliminary Product Bulletin

## APPLICATIONS

- Presentation
- Video Editing
- Video Authoring
- Video Teleconferencing
- Interactive Education
- Games

## FEATURES

- Direct ISA/MCA bus interface
- Interlaced or non-interlaced output
- Pixel clock rates up to 85 MHz
- Video inputs
  - 8:8:8 RGB at 40 MHz
  - (1)5:5:5 (T)RGB at 85 MHz
  - 5:6:5 RGB at 85 MHz
  - 4:2:2 YUV at 85 MHz
  - Tagged 4:2:2 YUV at 85 MHz
- Graphics input
  - 4-bit pseudo-color at 85 MHz
  - 8-bit pseudo-color at 85 MHz

(cont. next page)

## MediaDAC™

## OVERVIEW

The CL-PX2080 MediaDAC is a multiple-source, digital-to-analog video converter. It manages and mixes two different video data streams while converting the input data into the format of the display subsystem, and changes color space and resolution from the input to the output format in real-time.

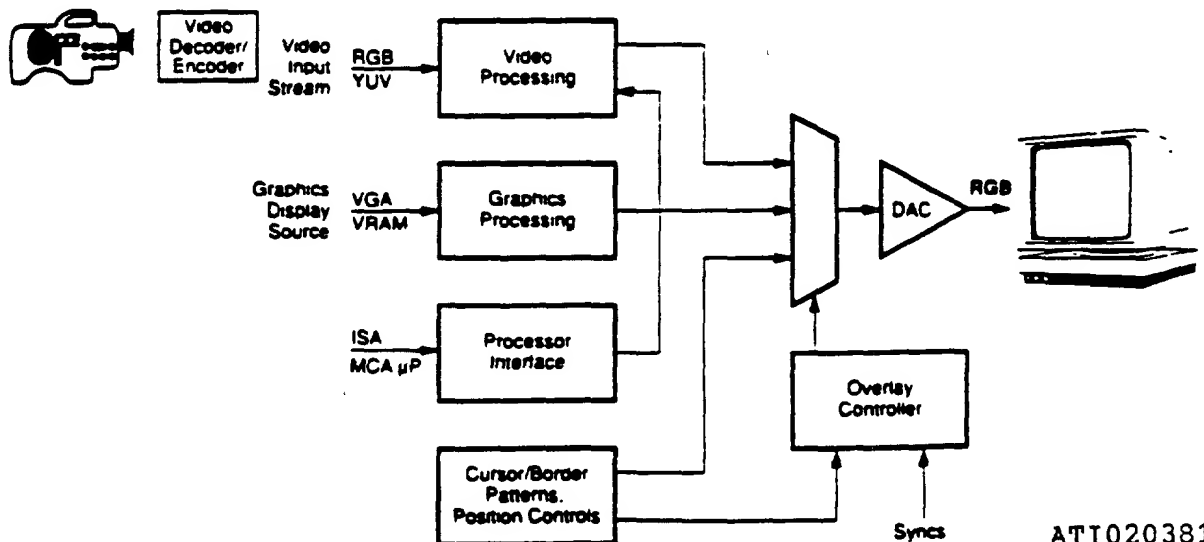
## ARCHITECTURAL OVERVIEW

This section describes the architecture and functionality of the CL-PX2080 MediaDAC.

The CL-PX2080 has a video port for YCbCr or RGB data and two graphics ports for both 8-bit VGA and 32-bit high-resolution ports. Its display functions include pseudo-color, display of true color RGB data, X-zooming of video port data, hardware cursor controls and a combination of three graphics overlay controls.

(cont. next page)

System Block Diagram



MAY CONTAIN  
CONFIDENTIAL INFORMATION  
SUBJECT TO PROTECTIVE ORDER

ATI020381

August 1992

CONFIDENTIAL-SUBJECT TO COURT PROTECTIVE ORDER  
LEVEL ONE Civil Action No. 952388R (AJB)

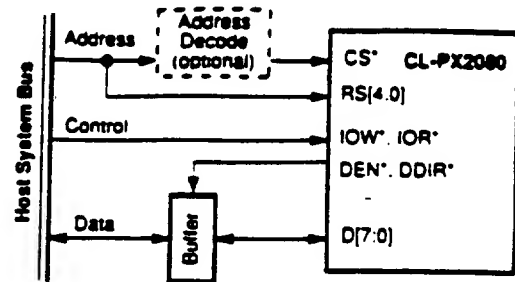




## FEATURES (cont.)

- 5:6:5 RGB at 85 MHz
- 5:5:5 RGB at 85 MHz
- 8:8:8 RGB at 40 MHz
- **Zoom controls**
- **Hardware cursor controls**
- **Three overlay combination controls**
  - Tagged chroma color key
  - Graphics overlay color key
  - X/Y window

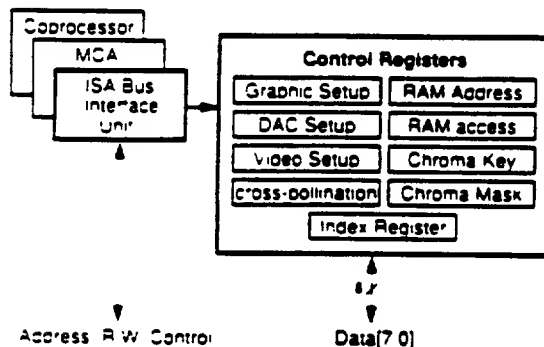
this trend with its local Hardware Interface Mode, illustrated below.



## ARCHITECTURAL OVERVIEW (cont.)

### Host System Interface

The CL-PX2080 connects directly to ISA and MCA buses, internally decoding a 16-bit address and responding as an 8-bit peripheral. Its internal ISA/MCA bus interface eliminates most of the costly glue circuitry common to many personal computer system expansion boards.



In response to customer demands for increased performance, the display subsystem in many new systems has migrated onto the host processor bus. The CL-PX2080 is designed to accommodate

### Video Input Interface

The video input interface accepts digitized video in a wide range of formats. The video data stream is converted to its final output format, then is overlaid with processed graphics data and cursor data.

#### Features:

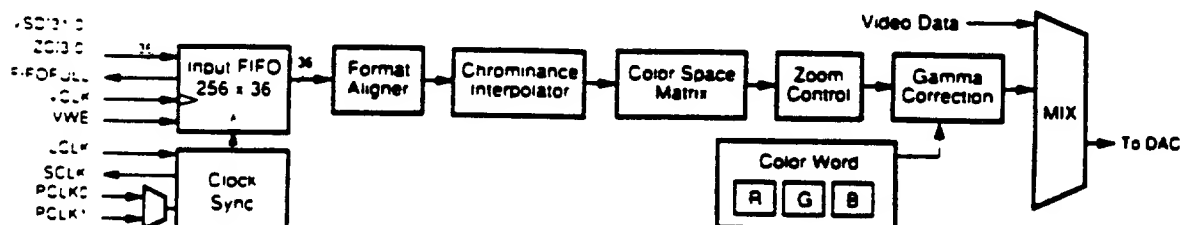
- 36-bit input data path (32-bit data, 4-bit zoom code)
- Internal 256 x 36-bit input FIFO that supports:
  - 24-bit RGB data (up to 40 mega-pixels per second)
  - 16-bit RGB or YCbCr data (up to 85 mega-pixels per second)
- Chrominance interpolation
- Color-space conversion
- Zoom control

### Graphics Frame Buffer Interface

The CL-PX2080 accepts data from the graphics display source through either of two paths:

- an 8-bit VGA data path, or
- a 32-bit VRAM serial data path.

### Video Input Interface



MAY CONTAIN  
CONFIDENTIAL INFORMATION  
SUBJECT TO PROTECTIVE ORDER

AT1020382

Both paths allow CL-PX2080-based, next-generation PC graphics subsystems to maintain compatibility with the large installed base of VGA systems and VGA-specific software, while also achieving higher performance and resolution via the VRAM serial data path.

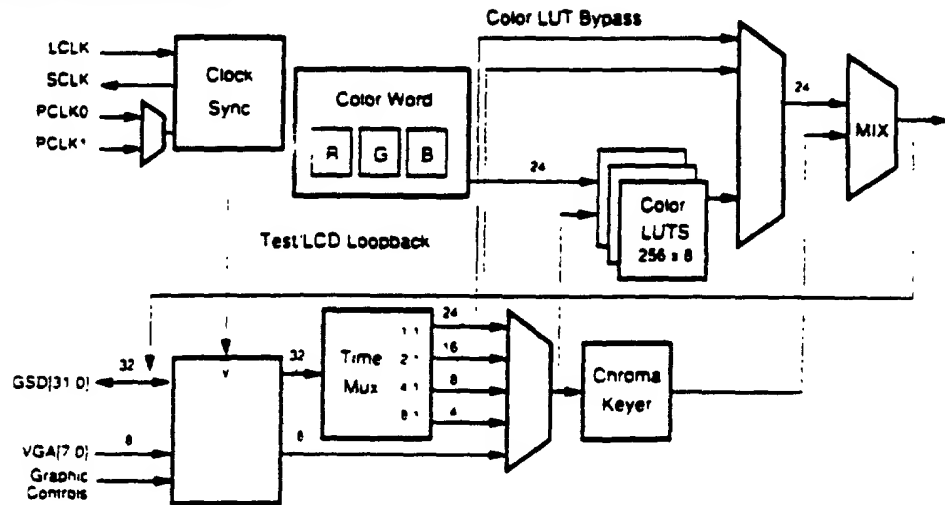
**Features:**

- VRAM Interface
  - 32-bit data bus

- Efficient pixel mapping within graphics-data-word
- Designed to accept data from VRAM serial ports, can be used with a variety of architectures

- VGA Interface
- True-color (CLUT bypass) option

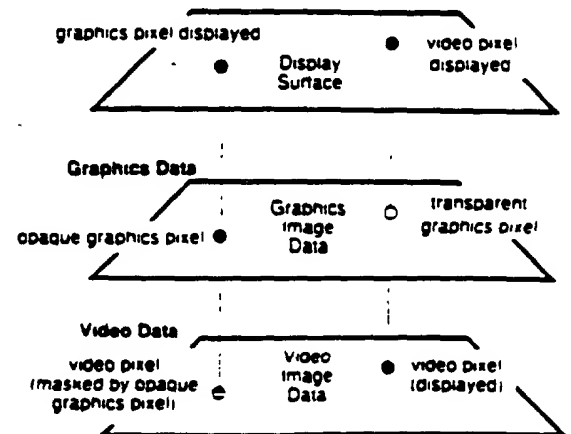
**Graphics Frame Buffer Interface**



**Graphics Overlay Control**

The graphics overlay controls allow a video image and a graphics image to be combined using a variety of operations (see figure at right).

Every graphics pixel is either transparent or opaque. The color information for an opaque pixel is displayed on the screen. The color information for a transparent pixel is not displayed; instead, the color information of the video pixel behind it is displayed on the screen. The graphics overlay controls determine which graphics pixels are transparent. The CL-PX2080 has 256 possible overlay combinations based on the video-pixel tag bit, the graphics-pixel overlay color, and the X/Y window of the video data.



MAY CONTAIN  
CONFIDENTIAL INFORMATION  
SUBJECT TO PROTECTIVE ORDER

ATI020383



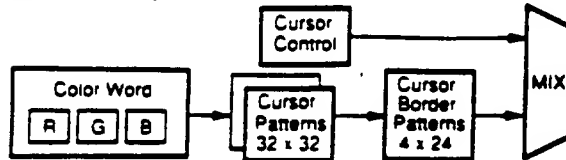
**Pixel  
Semiconductor**  
A Cirrus Logic Company

**CL-PX2080**  
*Preliminary Product Bulletin*

### Cursor

The CL-PX2080 implements an on-chip, three-color, user-definable hardware cursor in a 32 x 32 x 2-bit memory. This cursor works in both interlaced and non-interlaced systems.

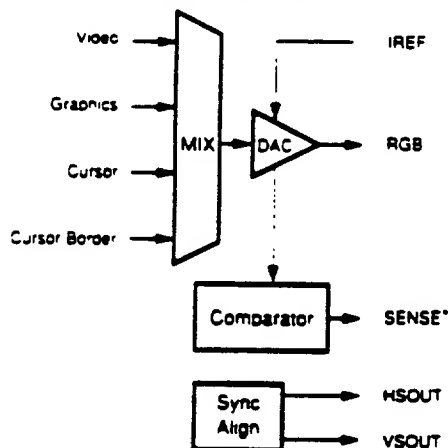
### Cursor Control



### Output DACs

The CL-PX2080 has three video-speed, 8-bit digital-to-analog converters, internal comparators to provide the sense function, and sync alignment logic. These form a complete RGB monitor interface.

### Source Mix and Monitor Interface



### Power-Down Mode

During the CL-PX2080 power-down condition, the DACs power-down and the RAM enters a low-power, data-retaining Standby Mode. The processor can read from or write to the RAM as long as the pixel clock is running. The RAM automatically powers-up during processor read/write cycles, and shuts down when processor access is completed. The three DAC-command registers are also accessible.

## Cirrus Logic Direct Sales Offices

### Domestic

**N CALIFORNIA**  
San Jose  
TEL 408/436-7110  
FAX 408/437-8960  
**S. CALIFORNIA**  
Tustin  
TEL 714/258-8303  
FAX 714/258-8307  
Thousand Oaks  
TEL 805/371-5381  
FAX 805/371-5382  
**ROCKY MT AREA**  
Denver, CO  
TEL 303/682-0050  
FAX 303/682-0053  
**S CENTRAL AREA**  
Austin TX  
TEL 512/794-8490  
FAX 512/794-8068

### Piano TX

TEL 214/965-2334  
FAX 214/964-3119  
**CENTRAL AREA**  
Chicago, IL  
TEL 708/490-5940  
FAX 708/490-5942  
**N EASTERN AREA**  
Andover MA  
TEL 508/474-9300  
FAX 508/474-9149  
New Brunswick, NJ  
TEL 908/603-7757  
FAX 908/603-7756  
**S EASTERN AREA**  
Boca Raton FL  
TEL 407/362-5225  
FAX 407/362-5235

### International

**JAPAN**  
Tokyo  
TEL 81/3-5389-5300  
FAX 81/3-5389-5540  
**SINGAPORE**  
TEL 65/3532122  
FAX 65/3532166  
**TAIWAN**  
Taipei  
TEL 886/2-718-4533  
FAX 886/2-718-4526  
**GERMANY**  
Munich  
TEL 49/8152-2030  
FAX 49/8152-6211  
**UNITED KINGDOM**  
Wotton-under-Edge  
TEL 44/0727-872424  
FAX 44/0727-875919

### The Company

Cirrus Logic, Inc., is a leading supplier of high-integration peripheral controller circuits for mass storage, graphics, and data communications. The company also produces state-of-the-art software and firmware to complement its product lines. Cirrus Logic technology is used in leading-edge personal computers, engineering workstations, and office automation.

Pixel Semiconductor, Inc., a subsidiary of Cirrus Logic, Inc., is a developer of integrated circuits for advanced display systems. These circuits enable the integration of real-time video with traditional computer graphics.

Cirrus Logic's fabless manufacturing strategy, unique in the semiconductor industry, employs a full manufacturing infrastructure to ensure maximum product quality, availability and value for our customers.

Talk to our systems and applications specialists, see how you can benefit from a new kind of semiconductor company.

© Copyright Pixel Semiconductor, Inc. 1982

Preliminary product information describes products that are in production, but for which full characterization data is not yet available. Pixel Semiconductor, Inc., believes that the information contained in this document is accurate and reliable. However, it is marked *Preliminary* and is subject to change without notice. Pixel Semiconductor, Inc., assumes no responsibility for its use nor for infringements of patents or other rights of third parties. This document implies no license under patents or copyrights. Trademarks in this document belong to their respective companies.

CIRRUS LOGIC, Inc.,  
3100 West Warren Ave  
Fremont, CA 94538

TEL 510/623-8300  
FAX 510/226-2180

752080-001

MAY CONTAIN  
CONFIDENTIAL INFORMATION  
SUBJECT TO PROTECTIVE ORDER

ATI020384

CONFIDENTIAL-SUBJECT TO COURT PROTECTIVE ORDER  
LEVEL ONE Civil Action No. 952388R (AJB)



Pixel  
Semiconductor  
A Cirrus Logic Company

CL-PX2080  
MediaDAC™  
Datashet

## FEATURES

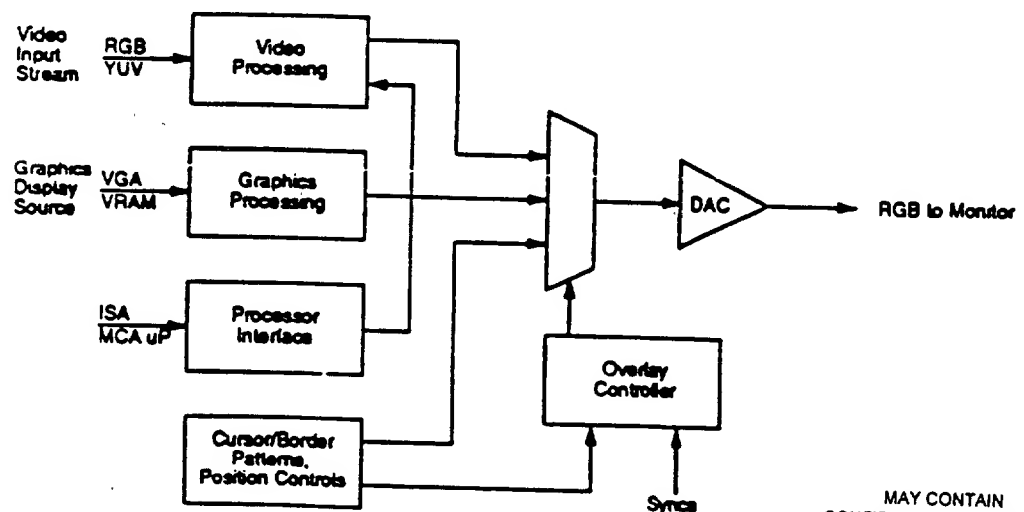
- CL-PX0070 superset
- direct ISA/MCA bus interface
- Interlaced or non-interlaced output
- up to 85 MHz pixel clock rates
- video inputs
  - 8:8:8 RGB at 40 MHz
  - (1)5:5:5 (T)RGB at 85 MHz
  - 5:6:5 RGB
  - 4:2:2 YUV at 85 MHz
  - tagged 4:2:2 YUV at 85 MHz
- zoom controls
- hardware cursor controls
- graphics input
  - 4-bit pseudocolor at 85 MHz
  - 8-bit pseudocolor at 85 MHz
  - 5:6:5 RGB at 80 MHz
  - (1)5:5:5 (T)RGB at 85 MHz
  - 8:8:8 αRGB at 40 MHz

MediaDAC™

- 3 overlay combination controls
  - tagged chroma color key
  - graphics overlay color key
  - XY window

## APPLICATIONS

- Presentation
- Video Editing
- MultiMedia Authoring
- Video Teleconferencing
- Animation
- Video capture and scaling for VSPs



MAY CONTAIN  
CONFIDENTIAL INFORMATION  
SUBJECT TO PROTECTIVE ORDER

ATI020370

CONFIDENTIAL  
CONFIDENTIAL-SUBJECT TO COURT PROTECTIVE ORDER  
LEVEL ONE Civil Action No. 952388R (AJB)

May 29, 1992



## OVERVIEW

The CL-PX2080 MediaDAC™ is a multiple-source video digital to analog converter. It manages and mixes two different video data streams while converting the input data into the format of the display subsystem, and changes color space and resolution from the input to the output format in real time.

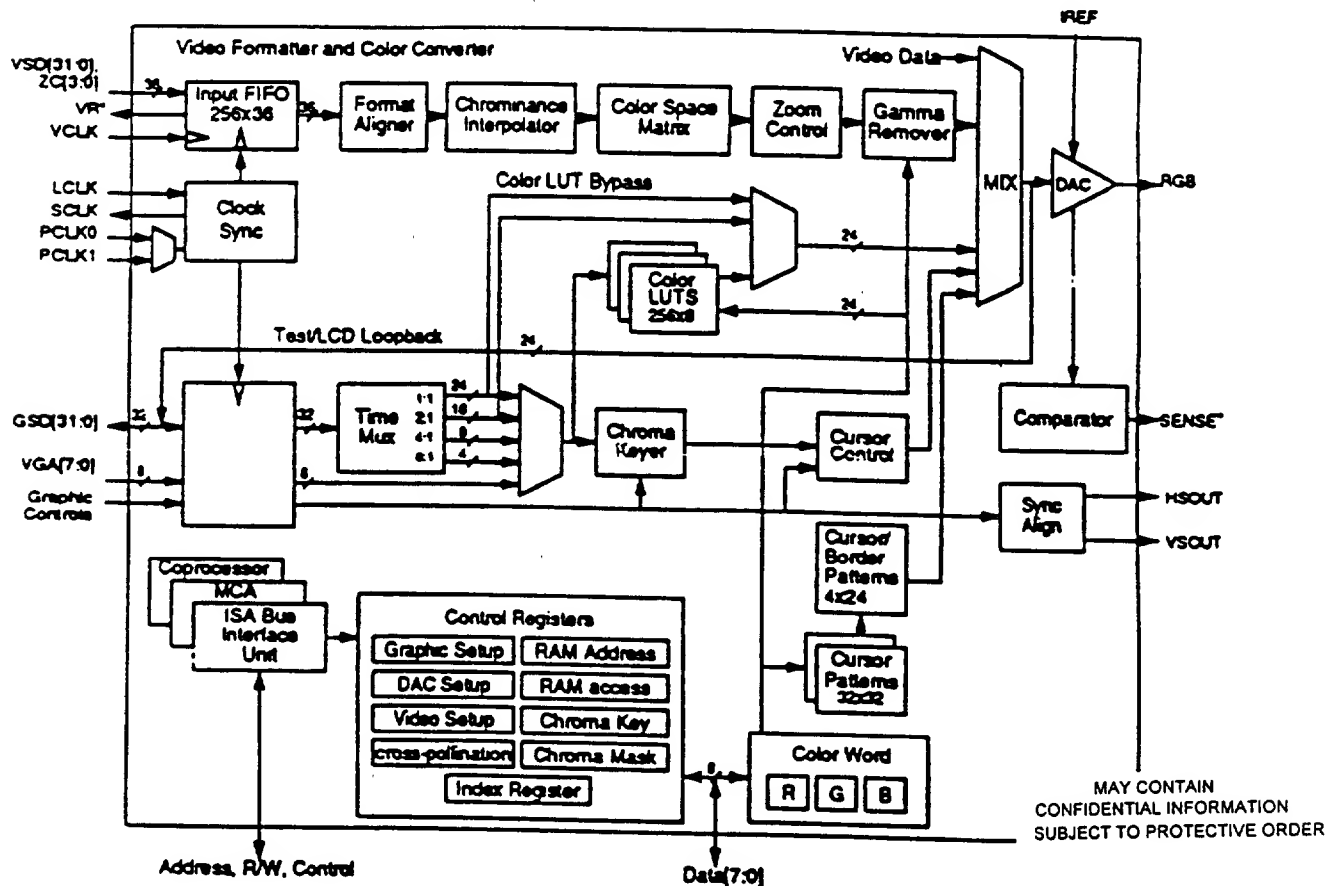
The CL-PX2080 has a video port for YCbCr or RGB data and a graphics port with both 8-bit VGA and 32-bit high resolution ports. Its display functions include:

- pseudocolor.
- display of true-color RGB data.
- X zooming.
- hardware cursor controls, and
- a combination of three graphics overlay controls.

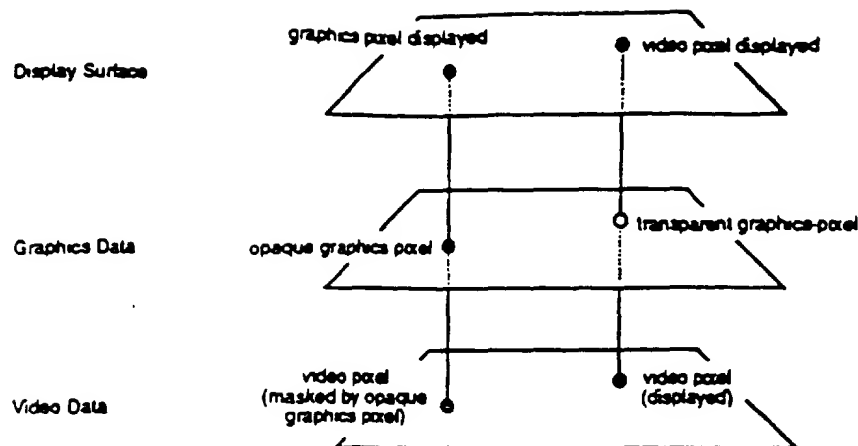
## Host System Interface

The CL-PX2080 connects directly to ISA and MCA buses, internally decoding a 16-bit address and responding as an 8-bit peripheral. Its internal ISA/MCA bus interface eliminates most of the costly "glue" circuitry common to many personal computer system expansion boards.

In response to customer demands for increased performance, the display subsystem in many new systems has migrated onto the host-processor bus. The CL-PX2080 is designed to accommodate this trend; in host bus mode, the host processor directly accesses CL-PX2080 registers with 16-bit I/O addresses.



ATI020371



### Video Input Interface

The video input interface accepts digitized video in a wide range of formats. The video data stream is converted to its final output format, then mixed and/or overlaid with processed graphics data and cursor data.

#### Features:

- 36-bit input data path;
- internal 256x36 bit input FIFO that supports:
  - 24-bit RGB data (up to 40 mega-pixels/s)
  - 16-bit RGB or YCbCr data (up to 80 mega-pixels/s)
- chrominance format alignment;
- color-space interpolation;
- zoom control.

### Graphics Frame Buffer Interface

The CL-PX2080 accepts data from the graphics display source through either of two paths:

- an 8-bit VGA data path, or
- a 32-bit VRAM serial data path.

Both paths allow CL-PX2080-based, next-generation PC graphics subsystems to maintain compatibility with the large installed base of VGA systems and VGA-specific software, while also achieving higher performance and resolution via the VRAM serial data path.

#### Features:

- VRAM Interface
  - 32-bit data bus
  - efficient pixel mapping within graphics-data word
- VGA Interface
- true color (CLUT bypass) option

### Cursor

The CL-PX2080 implements an on-chip, three-color, user-definable hardware cursor in a 32x32x2 bit memory. This cursor works in both interlaced and non-interlaced systems.

### Graphics Overlay Control

The graphics overlay controls allow a video image and a graphics image to be combined using a variety of operations (see above figure).

Every graphics pixel is either transparent or opaque. The color information for an opaque pixel is displayed on the screen. The color information for a transparent pixel is not displayed; instead, the color information of the video pixel behind it is displayed on the screen. The graphics overlay controls determine which graphics pixels are transparent.

### Output DACs

TBD

### Power Down Mode

During the CL-PX2080's power-down condition, the DACs power down and the RAM enters a low-power, data-retaining standby mode. The processor can read from or write to the RAM as long as the pixel clock is running. The RAM automatically powers up during processor read/write cycles, and shuts down when processor access is completed. The three DAC-command registers are also accessible.

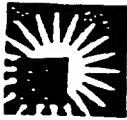
MAY CONTAIN  
CONFIDENTIAL INFORMATION  
SUBJECT TO PROTECTIVE ORDER

ATI020372

May 29, 1992

CONFIDENTIAL

CONFIDENTIAL-SUBJECT TO COURT PROTECTIVE ORDER  
LEVEL ONE Civil Action No. 952388R (AJB)



Pixel  
Semiconductor

CL-PX2080  
MediaDAC™

## Cirrus Logic Direct Sales Offices

### Domestic

N. CALIFORNIA  
San Jose  
TEL: 408/436-7110  
FAX: 408/437-8960

S. CALIFORNIA  
Tustin  
TEL: 714/258-8303  
FAX: 714/258-8307

Thousand Oaks  
TEL: 805/371-5381  
FAX: 805/371-5382

ROCKY MT AREA  
Boulder, CO  
TEL: 303/939-9739  
FAX: 303/440-5712

S. CENTRAL AREA  
Austin, TX  
TEL: 512/794-8490  
FAX: 512/794-8069

Plano, TX  
TEL: 214/985-2334  
FAX: 214/964-3119

N. EASTERN AREA  
Andover, MA  
TEL: 508/474-9300  
FAX: 508/474-9149

S. EASTERN AREA  
Boca Raton, FL  
TEL: 407/994-9883  
FAX: 407/994-9887

### International

JAPAN  
Tokyo  
TEL: 81/3-5389-5300  
FAX: 81/3-5389-5540

SINGAPORE  
TEL: 65/3532122  
FAX: 65/3532166

TAIWAN  
Taipei  
TEL: 886/2-718-4533  
FAX: 886/2-718-4526

GERMANY  
Herrsching  
TEL: 49/8152-2030  
FAX: 49/8152-6211

UNITED KINGDOM  
Hertfordshire, England  
TEL: 44/0727-872424  
FAX: 44/0727-875919

## The Company

Cirrus Logic, Inc., is a leading supplier of high-integration peripheral controller circuits for mass storage, graphics, and data communications. The company also produces state-of-the-art software and firmware to complement its product lines. Cirrus Logic technology is used in leading-edge personal computers, engineering workstations, and office automation.

Pixel Semiconductor, Inc., a subsidiary of Cirrus Logic, Inc., is a developer of integrated circuits for advanced display systems. These circuits enable the integration of real-time video with traditional computer graphics.

Cirrus Logic's extensive quality assurance program — one of the industry's most stringent — ensures the utmost in product reliability. Talk to our systems and applications specialists; see how you can benefit from a new kind of semiconductor company — Cirrus Logic.

© Copyright Pixel Semiconductor, Inc., 1992

Preliminary product information describes products which are in production, but for which full characterization data is not yet available. Pixel Semiconductor, Inc., believes that the information contained in this document is accurate and reliable. However, it is marked Preliminary and is subject to change without notice. Pixel Semiconductor, Inc. assumes no responsibility for its use, nor for infringements of patents or other rights of third parties. This document implies no license under patents or copyrights. Trademarks in this document belong to their respective companies.

CIRRUS LOGIC, Inc., 3100 West Warren Ave. Fremont, CA 94538  
TEL: 510/623-8300 FAX: 510/226-2160

752070-001

CONFIDENTIAL — May 29, 1992

ATI020373

CONFIDENTIAL-SUBJECT TO COURT PROTECTIVE ORDER  
LEVEL ONE Civil Action No. 952388R (AJB)

MAY CONTAIN  
CONFIDENTIAL INFORMATION  
SUBJECT TO PROTECTIVE ORDER